XSTAR

A Spectral Analysis Tool

Version 2.2

T.R. Kallman

NASA Goddard Space Flight Center

April 20, 2010

Contents

Chapter 1

Introduction

1. What is XSTAR?

XSTAR is a command-driven computer program for calculating the physical conditions and emission spectra of photoionized gases. It may be applied in a wide variety of astrophysical contexts. Stripped to essentials, its job may be described simply: A spherical gas shell surrounding a central source of ionizing radiation absorbs some of this radiation and reradiates it in other portions of the spectrum; XSTAR computes the effects on the gas of absorbing this energy, and the spectrum of reradiated light. In many cases other sources (or sinks) of heat may exist, for example, mechanical compression or expansion, or cosmic ray scattering. XSTAR permits consideration of these effects as well. The user supplies the shape and strength of the incident continuum, the elemental abundances in the gas, its density or pressure, and its thickness; the code returns the ionization balance and temperature, opacity, and emitted line and continuum fluxes. The solution divides into several distinct parts: transfer of the incident radiation into the cloud; calculation of the temperature, ionization, and atomic level populations at each point in the cloud; and transfer of the emitted radiation out of the cloud. XSTAR v2 is written in standard fortran77, and has been tested on a variety of unix platforms

2. Scope of This Document

The new user will need to first read Chapter 14 on how to obtain a copy of XSTAR, and is then advised to read Chapter 2 which gives the flavor of an XSTAR session, and provide sufficient information to get started. The chapters 3 and 4, give an overview of the code structure and describe the input commands. The output is described in Chapter 5. More experienced users may wish to read Appendix 9, which discusses the physical assumptions made in XSTAR, or Appendix 10, on the code structure.

We emphasize here and throughout this document that XSTAR version 2 represents an almost complete rewrite of version 1. This affects not only the internal operation of the code and the numerical results, but also the user interface. These changes were motivated primarily by the desire for a more streamlined operation, and for flexibility incorporating future and current improvements in atomic data. In addition, we have tried to learn from experience gained from version 1 by eliminating options which were seldom used in favor of clearer and more straightforward specification of input parameters, and we have replaced the code which parsed the input commands and which did not function as required in some situations. In order to do so we have borrowed heavily from the well developed input and output code, and installation scripts, which have been developed in this laboratory for the FTOOLS software package. This includes the use of FITS files for some output, As a consequence, it will be necessary for even experienced users of version 1 to become familiar with the new interface.

3. Acknowledgments

We would like to thank many colleagues for their suggestions, bug reports, and (occasionally) source code. The initial development of XSTAR was promoted by Prof. Dick McCray at the University of Colorado. Contributions to the code and atomic database have come from John Raymond, Barry Smith, Ian Stevens and Yuan-Kuen Ko. Much of the impetus for work on versions 1 and 2 came from Julian Krolik. The work for version 2 could not have been carried out without programming support and advice from Tom Bridgman, James Peachey, Bryan Irby, and Bill Pence. A great deal of crucial work on the atomic data was done by Manuel Bautista, and also by Patrick Palmeri, Claudio Mendoza, Javier Garcia, Mike Witthoeft and Ming-Feng Gu. The production of this manual and the circulation of the code has been funded by NASA through the Astrophysical Data Program, Grant NAG 5-1732.

4. Useful addresses

Timothy Kallman, Code 662	Internet:	tim@xstar.gsfc.nasa.gov
NASA GSFC	Telephone:	(301)-286-3680
Greenbelt MD 20771, USA	FAX:	(301)-286-1684

Chapter 2

A Walkthrough of XSTAR

In this section we run through a couple of fictitious interactive XSTAR sessions to illustrate how to use XSTAR's interface. First of all, it is important that the environment variables are set in a manner similar to that required by FTOOLS. This is described in more detail in Chapter 14. Once XSTAR is installed and configured, at the unix prompt, type:

```
unix> xstar
```

By invoking XSTAR with this simple command, you will be prompted for a series of physical and control parameters for the simulation.

The input parameters are: The model covering fraction, temperature, pressure or density, spectrum shape and ionizing luminosity, column density, ionization parameter, and element abundances relative to their cosmic abundances. Definitions for these and the units assumed are described in detail in chapter 4. All input parameters have default values, selected by pressing return at the prompt, thus it is possible to simply start XSTAR as above to invoke the default model.

Input of these parameters is handled through an IRAF-style interface, XPI, developed for the FTOOLS suite of programs. This has several features which may prove convenient to the user: (i) Parameter values from an invocation of XSTAR are stored and available as default values for the next XSTAR run. These are shown in square brackets when the user is prompted for values. (ii) Default values of input parameters are stored in the file xstar.par which is stored in the directory specified by the PFILES environment variable. (iii) Each parameter has an allowed range of values, shown in parentheses during the prompting. Input values outside of this range will result in exiting the program. (iv) Input parameters can be hidden from the prompting process. Such parameters, in the default xstar.par, are those which are expected to be unneeded for simple problems. Manipulation of these parameters requires a slightly more advanced familiarity with the code. (v) Parameter values can be input from the command line rather than by prompting. The value of this scripting capability will be seen in Chapter 7 which discusses building table models for the XSPEC spectral analysis program. All these features are described in more detail in the documentation for FTOOLS

`(c.f. <http://heasarc.gsfc.nasa.gov/docs/frames/hhp_sw.html>)`

It is also possible to run a model at a fixed temperature, i.e. to disable the thermal equilibrium condition. In order to do this the ordinarily hidden parameter **niter** must be set to 0, and then the **temperature** parameter must be set to the desired value (in units of 10^4 K).

In the sections that follow are a few XSTAR sessions to illustrate using the program. We illustrate both the prompted and the command line invocation of XSTAR with only the required parameters specified.

1. XSTAR Model: Spherical cloud

In this example we model a spherical, constant density cloud with a source at its center. The cloud is optically thin. The source luminosity is 10^{28} erg s^{-1}. The ionization parameter at the inner edge of the cloud is log(ξ)=5. The ionizing spectrum is a power law with energy index -1. We show how this model can be run in two ways: first by invoking XSTAR with no parameter values and utilizing the prompting for parameter values from XPI, and second by entering parameter values directly on the command line. In the former case, the prompt strings are more descriptive than the parameter values themselves, but the net result is the same in both cases.

Using prompting:

```
unix > xstar
 xstar version 2.2.0
covering fraction (0.:1.) [1.]
temperature (/10**4K) (0.:1.e4) [10000.]
constant pressure switch (1=yes, 0=no) (0:1) [0]
pressure (dyne/cm**2) (0.:1.) [0.03]
density (cm**-3) (0.:1.e18) [1.e+4]
spectrum type?[pow]
radiation temperature or alpha?[-1.]
luminosity (/10**38 erg/s) (0.:1.e10) [1.e-6]
column density (cm**-2) (0.:1.e25) [1.E17]
log(ionization parameter) (erg cm/s) (-10.:+10.) [5.]
hydrogen abundance (0.:100.) [1.]
helium abundance (0.:100.) [1.]
carbon abundance (0.:100.) [1]
nitrogen abundance (0.:100.) [1]
```

```
oxygen abundance (0.:100.) [1]
fluorine abundance (0.:100.) [1.0]
neon abundance (0.:100.) [1]
sodium abundance (0.:100.) [1.0]
magnesium abundance (0.:100.) [1]
aluminum abundance (0.:100.) [1.0]
silicon abundance (0.:100.) [1]
phosphorus abundance (0.:100.) [1.0]
sulfur abundance (0.:100.) [1]
chlorine abundance (0.:100.) [1.0]
argon abundance (0.:100.) [1]
potassium abundance (0.:100.) [1.0]
calcium abundance (0.:100.) [1]
scandium abundance (0.:100.) [1.0]
titanium abundance (0.:100.) [1.0]
vanadium abundance (0.:100.) [1]
chromium abundance (0.:100.) [1.0]
manganese abundance (0.:100.) [1.0]
iron abundance (0.:100.) [1]
cobalt abundance (0.:100.) [1.0]
nickel abundance (0.:100.) [1]
copper abundance (0.:100.) [1.0]
zinc abundance (0.:100.) [1.0]
model name[filled sphere]
```

Using the command line:

```
xstar cfrac=1 temperature=1000. pressure=0.03 density=1.E+4 spectrum='pow'
```

```
trad=-1. rlrad38=1.E-16  column=1.E+16   rlogxi=5.  lcpres=0 habund=1 heabund=1
 liabund=0. beabund=0. babund=0. cabund=1. nabund=1 oabund=1 fabund=1 neabund=1
naabund=1 mgabund=1 alabund=1 siabund=1 pabund=1 sabund=1 clabund=1 arabund=1
kabund=1 caabund=1 scabund=1 tiabund=1 vabund=1 crabund=1 mnabund=1 feabund=1
coabund=1 niabund=1 cuabund=1 znabund=1 modelname='filled sphere' niter=0  npass=1
critf=1.E-07 nsteps=6 xeemin=0.04 emult=0.5 taumax=5. lprint=1  ncn2=999
radexp=0 vturb=1.
```

2. xstar Model: H II region

In this example we model an H II region with parameters corresponding to the first of the Lexington benchmarks (see appendix C). In this case we give only the prompted values. This illustrates the use of a blackbody spectrum, with temperature given in keV, and non-solar abundances.

```
unix > xstar
 xstar version 2.2
covering fraction (0.:1.) [1.]
temperature (/10**4K) (0.:1.e4) [100.] 1.
constant pressure switch (1=yes, 0=no) (0:1) [0]
pressure (dyne/cm**2) (0.:1.) [0.03]
density (cm**-3) (0.:1.e18) [1.E+12] 1.e+2
spectrum type?[pow] bbody
radiation temperature or alpha?[-1.] 0.004
luminosity (/10**38 erg/s) (0.:1.e10) [1.] 12.7
column density (cm**-2) (0.:1.e25) [1.E23]
log(ionization parameter) (erg cm/s) (-10.:+10.) [2.] 0.15
hydrogen abundance (0.:100.) [1.]
```

```
helium abundance (0.:100.) [1.]

carbon abundance (0.:100.) [1.]0.5945

nitrogen abundance (0.:100.) [1.] 0.3639

oxygen abundance (0.:100.) [1.] 0.4739

neon abundance (0.:100.) [1.] 1.7865

magnesium abundance (0.:100.) [1.] 0.

silicon abundance (0.:100.) [1.] 0.

sulfur abundance (0.:100.) [1.] 0.563

argon abundance (0.:100.) [1.] 0.

calcium abundance (0.:100.) [1.] 0.

iron abundance (0.:100.) [1.] 0.

nickel abundance (0.:100.) [1.] 0.

model name[xstar Default] H II region
```

3. xstar Model: Quasar Broad Line Cloud

In this example we model a quasar broad line cloud. In this case we give only the prompted values. This illustrates the use of constant pressure and a power law spectrum, with spectral index input in energy units.

```
unix > xstar

 xstar version 2.2

covering fraction (0.:1.) [1.] 0.

temperature (/10**4K) (0.:1.e4) [100.] 1.

constant pressure switch (1=yes, 0=no) (0:1) [0] 1

pressure (dyne/cm**2) (0.:1.) [0.03]

density (cm**-3) (0.:1.e18) [1.E+12] 1.e+10
```

```
spectrum type?[pow] pow

radiation temperature or alpha?[-1.] -0.9

luminosity (/10**38 erg/s) (0.:1.e10) [1.] 1.e+8

column density (cm**-2) (0.:1.e25) [1.E23]

log(ionization parameter) (erg cm/s) (-10.:+10.) [2.] 0.2

hydrogen abundance (0.:100.) [1.]

helium abundance (0.:100.) [1.]

carbon abundance (0.:100.) [1.]

nitrogen abundance (0.:100.) [1.]

oxygen abundance (0.:100.) [1.]

neon abundance (0.:100.) [1.]

magnesium abundance (0.:100.) [1.]

silicon abundance (0.:100.) [1.]

sulfur abundance (0.:100.) [1.]

argon abundance (0.:100.) [1.]

calcium abundance (0.:100.) [1.]

iron abundance (0.:100.) [1.]

nickel abundance (0.:100.) [1.]

model name[xstar Default] quasar BLR cloud
```

Chapter 3

An Overview of XSTAR

1. Basic Operation

An XSTAR session consists of several basic steps: initial setup, model calculation, and final printout. The initial setup consists of the input of various parameter values necessary to specifying a model, reading in atomic data files, and the program's internal initialization. The model calculation consists of the calculation of ionization, excitation, and thermal equilibrium, and radiation transfer (as described in Chapter 9) at each of a number of spatial grid points. After the model calculation various quantities are output, and the program terminates. The details of the input parameters are described in Chapter 4, and the output is described in Chapter **??**. Note that both of these have changed significantly since version 1. In this chapter we describe general features of the XSTAR operation which are important to the user.

1.1. Important note

Although XSTAR is designed to be relatively user-friendly, and is supplied along with user-oriented software for data analysis, its use inevitably implies risk of incorrect or undesired results. Common among these are application to situations where the physical assumptions or numerical accuracy breaks down, or attempts to solve problems beyond the capabilities of the machine being used. The user is urged to read chapter 6, 'Problems and Pitfalls', before attempting changes to parameter values far beyond those in the sample parameter files.

1.2. Command syntax

As described in the previous chapter, XSTAR is invoked from the command line either with or without parameter specifications. In the latter case the user is prompted for parameter values. Operation can be interrupted or invoked in the background.

1.3. Defining input spectra

XSTAR allows users to input ionizing spectra constructed of several simple functional forms: bremsstrahlung, blackbody, or power law. Alternatively, more complicated spectra can be stored in an ascii file, and used by XSTAR as input. This is discussed in more detail in Chapter 4.

1.4. FITS Files & Graphical Output

The primary output method for XSTAR is via FITS files. The Interactive Data Language (IDL) has a number of routines designed for reading data from FITS files (see http://idlastro.gsfc.nasa.gov). In addition, the program FV supported by the HEASARC provides viewing and plotting capabilities of FITS data. The CFITSIO package provides additional support to read FITS files in both C and Fortran programming languages. Information on these packages is available through the 'Software' link on the HEASARC home page (http://heasarc.gsfc.nasa.gov/).

1.5. The Atomic Database

The atomic data is supplied with the code in a single binary FITS file named atbd.fits and known as the Atomic Database. An ascii version of this is available from the xstar website.

1.6. Supported Platforms

An additional benefit of integrating XSTAR into the FTOOLS suite is that support is maintained across a wide range of platforms, notbly Linux (x86, redhat) and Mac OSX.

1.7. Installation

Version 2 is distributed by default as part of the HEAdas analysis package. It is also available to be installed and run stand-alone, and the procedures for obtaining and installing it are described in Chapter 14.

1.8. Subroutine XSTAR

One consequence of the change in the user interface between versions 1 and 2 was a loss of certain features, and some flexibility in specifying input parameters. Examples include pipelined models and variable gas density. This sacrifice was deemed acceptable in the interests of streamlined and simplified execution, and because these features were not often used (as far as we know). In an effort to preserve these features in some form for dedicated users we also provide the capablility to call XSTAR as a fortran subroutine, which allows the user flexibility in specifying geometry, spectrum and density. This is described in Chapter 14.

2. Limitations

One of the potential pitfalls of using a 'blackbox' code such as XSTAR is the application to problems for which its inherent assumptions or numerics are invalid or inaccurate. The best way to avoid this is to thoroughly understand the computational methods as outlined in Appendix 9. The user is urged to read Chapter 6: "Problems and Pitfalls" in order to avoid the most obvious of these. In what follows we have attempted to supply a few simple rules, which will, we hope, keep the less careful user from going into forbidden territory.

2.1. Temperature

At high temperature, $\geq 10^9 K$, the assumption of non-relativistic electron velocities becomes invalid, and a variety of physical processes such as electron-positron pair production may come into play. At very low temperatures, $\leq 3000K$, many of the analytic fitting formulae used to parameterize atomic rates are unreliable, and the neglect of molecule formation and atomic fine structure cooling renders rates inaccurate. The code does not allow temperatures less than 3000K for this reason and because serious numerical errors can occur. If lower values are attempted the temperature is artificially reset to 3000K. On the high temperatures end, there is no mechanism preventing values greater than $10^9 K$, but the user should be aware that in this regime the rates are probably unreliable.

2.2. Density

The atomic rates for H and He-like ions are accurate for densities up to 10^{18} cm^{-3} by taking into account the effects of 3-body recombination and lowering of the continuum (Bautista et al. 1998, Bautista et al. 1999). However, these effects are not treated for other species which reduce the accuracy of the model results for these ions at high density. The density for which these effects become important for a given ion increases rapidly with effective charge of the ion, starting at about 10^{12} cm^{-3} for z=1 and at about 10^{16} cm^{-3} for z=8. At low densities numerical errors can occur; the user is urged to read chapter 6.

2.3. Column Density

For column densities greater than 1.5×10^{24} cm^{-2}, corresponding to a Thomson depth of unity, neglect of 'Comptonization', spectral modification arising from Compton scattering, makes model results incorrect.

In addition, the upper bound on volume density declines as the total column density increases. The reason is that the number of transitions in detailed balance increases rapidly as the product of volume density and column density rises. Once many transitions are in detailed balance, the thermal balance becomes determined by transitions involving rarer species and excited states which are not included in the calculation. As a rough rule of thumb, the product of volume density and column density should not exceed 10^{34} cm^{-5}.

3. Getting help

Sections of this manual are available using the *fhelp* command. Do not hesitate to contact the author with comments and questions.

Chapter 4

User Input to XSTAR

On invoking XSTAR from the command line, the user is prompted for a series of input values which are described below. These parameters are stored in a parameter file xstar.par, which must live in the directory specified by the user's PFILES environment variable. Upon the initial invocation of XSTAR, the default version of the parameter file is copied into the pfiles directory. Each parameter has associated with it a prompt string and a parameter name. The former is used when values are input via prompting, and the latter when values are input via the command line. In sections 1 – 3 we describe the parameters used for most straightforward applications of XSTAR. In section 5 we discuss the parameters which are ordinarily hidden from prompting, and which provide added flexibility.

1. Input Parameter Summary

Here we list the XSTAR parameters in entry order.

prompt string	parameter name	description
Covering Fraction	cfrac	Covering Fraction
Temperature	temperature	Gas Temperature ($10^4 K$)
Constant Pressure Switch	lcpres	1=yes, 0=no
Pressure	pressure	Define model pressure in dynes/cm^2
Density	density	Density in cm^{-3}
Spectrum	spectrum	Define the input spectrum
Spectrum File	spectrum_file	Name of spectrum file
Spectrum Units	spectun	0=energy, 1=photons
Radiation Temperature or Alpha	trad	keV or unitless (E^{α})
Luminosity	rlrad38	Luminosity in 10^{38} ergs/s
Column Density	column	Column density in cm^{-2}
log(ionization parameter)	rlogxi	log(ξ) (erg cm/s) or log(Ξ)
Hydrogen abundance	habund	Hydrogen Abundance relative to Solar
Helium abundance	heabund	Helium Abundance relative to Solar
Lithium abundance	liabund	Lithium Abundance relative to Solar
Beryllium abundance	beabund	Beryllium Abundance relative to Solar
Boron abundance	babund	Boron Abundance relative to Solar
Carbon abundance	cabund	Carbon Abundance relative to Solar
Nitrogen abundance	nabund	Nitrogen Abundance relative to Solar
Oxygen abundance	oabund	Oxygen Abundance relative to Solar
Fluorind abundance	fabund	Fluorine Abundance relative to Solar
Neon abundance	neabund	Neon Abundance relative to Solar
Magnesium abundance	mgabund	Magnesium Abundance relative to Solar
Aluminum abundance	alabund	Aluminum Abundance relative to Solar
Silicon abundance	siabund	Silicon Abundance relative to Solar
Phosphorus abundance	pabund	Phosphorus Abundance relative to Solar
Sulfur abundance	sabund	Sulfur Abundance relative to Solar
Chlorine abundance	clabund	Chlorine Abundance relative to Solar
Argon abundance	arabund	Argon Abundance relative to Solar
Potassium abundance	kabund	Potassium Abundance relative to Solar
Scandium abundance	scabund	Scandium Abundance relative to Solar
Titanium abundance	tiabund	Titanium Abundance relative to Solar
Vanadium abundance	vabund	Vanadium Abundance relative to Solar
Chromium abundance	crabund	Chromium Abundance relative to Solar
Manganese abundance	mnabund	Mangnese Abundance relative to Solar
Iron abundance	feabund	Iron Abundance relative to Solar
Cobalt abundance	coabund	Cobalt Abundance relative to Solar
Nickel abundance	niabund	Nickel Abundance relative to Solar
Zinc abundance	znabund	Zinc Abundance relative to Solar
Copper abundance	cuabund	Copper Abundance relative to Solar
Model Name	modelname	Name used for identifying this run

2. Description of XSTAR Parameters

In this section detailed descriptions are given for each of the input parameters in the order of entry.

2.1. Covering Fraction (cfrac)

This parameter determines whether the geometry is a complete sphere or covers only part of the continuum source. In the former case, photons escaping the cloud in the 'inward' direction are assumed to reenter the cloud at the inner edge owing to the assumption of spherical symmetry. Default is 1.0.

2.2. Temperature (temperature)

Define temperature, in units of $10^4 K$. If the parameter `niter` is set to 0 then the temperature is fixed at this value. Otherwise the value is used as a first guess in calculating the thermal equilibrium value. If the pressure is specified it is also used to calculate an initial guess at the gas density, $n = P/(kT)$, which is then used to calculate $\Delta R_{max} = N/n$. Default value is 1.

2.3. Constant Pressure Switch (lcpres)

This parameter chooses between constant density (value 0) and constant pressure (value 1). If the pressure is constant then the appropriate definition of ionization parameter (Ξ) is adopted. If the density is constant then the appropriate definition of ionization parameter (ξ) is adopted. The constant density case is the default.

2.4. Pressure (pressure)

Define model pressure in dynes cm^{-2}. Note that this quantity represents the full isotropic pressure (neutral atoms + ions + electrons + trapped line radiation) instead of just the pressure due to hydrogen atoms and protons. Whether or not this quantity is held fixed is determined by the value of the constant pressure switch lcpres. If the pressure is constant then the appropriate definition of ionization parameter (Ξ) is adopted. In the constant density case, lcpres=0, then this quantity is ignored.

2.5. Density (density)

Define model gas density, n. This is actually the hydrogen nucleus density, so that, e.g., the total particle density in a fully-ionized plasma with solar abundances is $2.3n$. Units are cm^{-3}. The default value is 1 cm^{-3}. Whether or not this quantity is held fixed is determined by the value of the constant pressure switch lcpres. If the density is constant then the appropriate definition of ionization parameter (ξ) is adopted. The constant density case is the default.

2.6. Spectrum (spectrum)

Define Spectrum. Choices and formats are similar to those used by XSPEC and are given in Section 3. Note that numerical problems can arise if the radiation field is zero throughout a significant energy range, because then the photoionization rates for some ions may be zero, and these appear in the denominator of the equations for the ionization balance. If your input spectrum is not one of the internally supported models, enter 'file' to specify a custom spectrum in a text file using the next two parameters.

2.7. Spectrum File (spectrum_file)

If the 'file' option is chosen for the spectrum type, you must provide a text file of the spectrum in your current working directory. The first line of the text file must be the number of energies listed in the table. The remaining lines are the energy channel (in eV) and the flux in units of photons cm^{-2} s^{-1} erg^{-1} or erg cm^{-2} s^{-1} erg^{-1} (see next subsection). Note that XSTAR will appropriately renormalize the luminosity. Also, be aware that small numbers (less than 1.e-30, say), may result in undesired results owing to the limitations of many machines in the range of exponents. Again, remember that it is not necessary to use actual physical units in specifying the input spectrum (except to distinguish between photon and energy fluxes) since the entire spectrum is renormalized to conform to the luminosity specified.

2.8. Spectrum Units (spectun)

The appropriate units for the spectrum file specified above (1=photons cm^{-2} s^{-1} erg^{-1}, 0=erg cm^{-2} s^{-1} erg^{-1}). Default is 0

2.9. Radiation Temperature or Alpha (trad)

This parameter pulls double duty, used to enter the radiation temperature in units of 10^7K in the case of a blackbody or in units of keV for bremsstrahlung input model. It also is used to input the power-law index *(in energy)*, α, in the case of a power-law model. Note that α is defined as in $L_\varepsilon \sim \varepsilon^\alpha$ so generally α will be *less* than zero (this is the opposite of the convention used by xspec). XSTAR always works with specific luminosity L_ε in units erg s^{-1} erg^{-1}, and never uses $\varepsilon L_\varepsilon$ or νF_ν, for example.

2.10. Luminosity (luminosity)

Define model luminosity integrated between 1 and 1000 Ry. Units are 10^{38} erg s^{-1}. Default value is 1.

2.11. column_density

Define model column density, N. Units are cm^{-2}. This quantity is used in calculation of thickness of model slab according to $\Delta R_{max} = N/n$ where n is the density or an estimate based on pressure and temperature initial values. The default value for N is 10^{21} cm^{-2}. The model calculation terminates when this value is reached.

2.12. log of the ionization_parameter= log(ξ) or log(Ξ) (rlogxi)

Define initial value of the log (base 10) of the model ionization parameter. If the density is held constant, the Tarter Tucker and Salpeter 1969 form is used: $\xi = L/(nR^2)$. If the pressure is held constant, a version of the Krolik, McKee, and Tarter (1981) form is used: $\Xi = L/(4\pi c R^2 P)$. Note that this differs from the original form by using the full isotropic pressure (neutral atoms + ions + electrons + trapped line radiation) instead of just the pressure due to hydrogen atoms and protons. This quantity is used in calculating the radius of the innermost edge of the shell by inverting the parameter definition.

2.13. Abundances

Atomic abundances are entered *relative to solar abundances* as defined in Grevesse, Noels and Sauval 1996, with 1.0 being defined as the solar value and the current default. Currently supported range for these values is $(0.0 \ldots 100.0)$.

The atomic species are all elements up to and including copper (Z=30). Note that as of version 2.2.0 many of these are treated using scaled hydrogenic atomic rate data.

2.14. Model Name (modelname)

Define model name, an 80 character string.

3. The XSTAR Input Spectral Models

This part of the manual provides more information on specific installed XSTAR input spectra, as well as the use of the 'user' defined model facility. All spectral shapes are given in terms of the energy flux per unit frequency interval.

3.1. Summary

bbody: A Black Body spectrum.

bremss: Thermal bremsstrahlung.

powerlaw: Simple photon power law.

file: Specify a custom spectrum in a text file (see the spectrum_file and spectun parameters).

The detailed description of these models are in the sections below.

3.2. bbody

A black body spectrum.

$$A(\varepsilon) = \varepsilon^3/(\exp(\varepsilon/kT) - 1) \tag{4.1}$$

where T is temperature in units of 10^7 K.

3.3. bremss

Thermal bremsstrahlung spectrum, including gaunt factors, but not including $e - e$ bremsstrahlung. The input parameter for this model is plasma temperature in keV .

3.4. powerlaw

Simple photon power law.

$$A(\varepsilon) = \varepsilon^{\alpha} \tag{4.2}$$

where α is the energy index of power law.

The user is cautioned that simple power laws can have unintended consequences owing to the fact that they are automatically extrapolated to the lowest (0.1 eV) and highest (1 MeV) energies employed in the calculation. This can cause processes such as stimulated recombination and Compton cooling to dominate the model results, and may not represent a physically realistic result. These effects can be avoided by the use of a simple file spectrum as demonstrated below.

3.5. File

Fluxes and energies are read from a text file, with name given by the 'spectrum_file' keyword. The format of the file is as follows: the first line must contain the integer number of (energy, flux) pairs; each of the the remaining lines contains one (energy, flux) pair, with energy in eV and flux in energy units (overall normalization is arbitrary). These values will be interpolated onto the energy grid used internally by XSTAR using logarithmic interpolation. An example of a file which results in a ε^{-1} spectrum power law spectrum between 0.1 Ry and 1000 Ry (and zero elsewhere) is as follows:

```
006
1.e-3  1.e-10
1.359 1.e-10
1.3598 1.e+11
1.3598e+5 1.e+6
1.360e+5 1.e-10
2.e+5  1.e-10
```

As already states, be aware that small numbers (less than 1.e-30, say), may result in undesired results owing to the limitations of many machines in the range of exponents. And remember that it is not necessary to use actual physical units in specifying the input spectrum (except to distinguish between photon and energy fluxes) since the entire spectrum is renormalized to conform to the luminosity specified.

4. Direct Editing of the Parameter File

While XSTAR will prompt users for the standard parameters, there are instances where you may wish to edit the parameter list directly. These can be edited directly with a text editor such as Emacs or vi. There are also several FTOOL parameter manipulation utilities: PSET, PGET, PUNLEARN, PLIST, & PQUERY, in addition to the GUI tool FLAUNCH.

Note that while there are comments recorded in the parameter file as originally defined in the FTOOLS build, modifying these parameters through the regular XSTAR prompts will cause this parameter file to be rewritten *without* the comments included.

5. Hidden Parameters

The input parameters described so far are those which are the most physically relevant to the xstar results. Other parameters which are more related to control of the computation are ordimarily hidden from prompting. Changing these parameters requires a understanding of the operation of the code. Note also that the parameter file formalism allows these parameters to be prompted along with the others, simply by changing the 'h' to an 'a' in the third field of the respective line of the parameter file.

5.1. Number of Steps (nsteps)

This parameter controls the maximum number of spatial zones used in a calculation, only in the case where the Courant condition step is larger than the size of the slab. That is, the step size is calculated as:

$$\Delta R = min(\mathrm{emult}/\kappa_m ax(\varepsilon), R/\mathrm{nsteps})$$

where emult is defined below, and $\kappa_m ax(\varepsilon)$ is the maximum opacity from the previous step calculation. $\kappa_m ax(\varepsilon)$ is only calculated from energies where the optical depth to the illuminated face of the cloud is less than taumax, where taumax is defined below. The default value for nsteps is 2.

5.2. write_switch (lwrite)

If the argument is a non-zero integer, causes level populations and line emissivities in the interior of the shell to be written to a fits dataset at each spatial step for later examination or plotting. These files are named 'xout_detail.lis' and 'xout_detal2.lis', and can become quite large ($\geq$ 10 Mb) for a model with many spatial zones. Various fits file manipulation routines can be used to filter and plot the quantities in these files.

Default value is 0.

5.3. print_switch (lprint)

This enables the printing of many quantities which are defined locally at the last spatial zone of the calculation. These include ion fractions, heating and cooling rates, line and continuum emissivities and opacitites, execution times, and level populations. They are printed to the log file, xout_step.log. Default value is 0, and results in just the 500 brightest line luminosities (sorted by luminosity) and depths, and the 500 brightest recombination continuum (sorted by luminosity) luminosities and depths to be printed to the log file at the end of the run. With a value 1 many additional quantities are printed, including ascii tables of continuum luminosities and depths, all line luminosities, along with all useful local quantities such as ion fractions, thermal rates, level populations. Note that these local quantities are printed only at the final step of the model run. With a value of 2 all rates affecting level populations, ionization, heating and cooling internal to the code are also printed. These require significant familiarity with the code. This switch affects only the ascii log file, xout_step.log. The standard fits output files are unaffected by the value of lprint.

5.4. Courant Multiplier (emult)

This is a constant factor used in calculating the spatial step size. For each radial zone, the size of the next zone is chosen to be

$$\Delta R = min(\text{emult}/\kappa_m ax(\varepsilon), R/\text{nsteps})$$

where emult is defined below, and $\kappa_m ax(\varepsilon)$ is the maximum opacity from the previous step calculation. $\kappa_m ax(\varepsilon)$ is only calculated from energies where the optical depth to the illuminated face of the cloud is less than taumax, where taumax is defined below. The default value for nsteps is 2. The default value of emult is 0.5, and values outside the range 0.1 – 1 are unlikely to be of any practical value.

5.5. Max Tau for Courant Step (taumax)

This quantity is used in calculating the spatial step size, as described in the previous subsection. Energy bins with continuum optical depth to the illuminated cloud face greater than taumax are not used when searching for the maximum photoelectric opacity. Default value is 5.

5.6. Min Electron Abundance (xeemin)

This is the minimum allowed electron fractional abundance. If the electron fraction falls below this value the current pass is ended. The default value is 0.01.

5.7. Ion Abundance Criterion for Multilevel Calculation (critf)

Ions whose abundance relative to total hydrogen (H I + H II) are less than this value after the preliminary ion abundance calculation (See Chapter 10) are not included in full multilevel calculation. The default value is 10^{-8}. This parameter should be changed with caution owing to the fact that it determines the size of the matrix that xstar tries to solve in calculating the level populations.

In versions of xstar 2.1lxx and earlier this matrix had a maximum size of 2400, and an attempt to solve for more than this number of levels simultaneously would result in xstar stopping with a message 'ipmat too large'.

In version 2.2 this limitation has been removed, and arbitrarily small values of critf can be accomodated. Also, a faster algorithm has been adopted for the multi-level calculation, so the speed advantages of large critf have been reduced. Also, the input parameter critf now refers to the fractional ion abundance (i.e. relative to the parent element) rather than the absolute (i.e. relative to H) ion abundance.

5.8. Turbulent Velocity (vturbi)

This parameter allows extra line broadening to be introduced into the calculation of the synthetic spectrum. The value is in km s^{-1}, and the line shapes are assumed to be Gaussian. If a small value is input, then the broadening is assumed to be the greater of vturbi and the local thermal velocity of the absorbing ion.

5.9. Number of Passes (npass)

This parameter determines how many complete calculations of the temperature and ionization structure of the model shell are made. Multiple passes are needed because there is no a priori knowledge of the optical depth of the shell in all the lines and continua, and these can affect the state of the gas in the interior of the shell. During the first pass the calculation proceeds through the shell, and assumes that all optical depths from points within the shell to the far edge of the shell are 0. If an integer greater than 1 is supplied as a parameter, XSTAR performs that number of iterations through the entire calculation, setting the optical depths to the far edge at the values calculated in the previous iteration. The odd numbered passes are made from the smallest to largest radius, while the even numbered passes are made in the inward direction. The emergent spectrum is not calculated accurately during the inward passes, so npass must be odd. Multi-pass calculations substantially improve the accuracy of the predictions made for shells with finite thickness, but they are much more time consuming than single-pass calculations. They also make use of temporary unformatted datasets, named 'xout_tmp.lis', 'xout_tmp2.lis', which can become quite large. The default value for this parameter is 1.

5.10. Number of Iterations (niter)

Set maximum number of iterations for thermal equilibrium and charge neutrality calculation at each spatial step. If this quantity is set to zero then a constant temperature run will result, and charge neutrality will not be calculated. If this quantity is negative, then charge neutrality will be calculated, but thermal equilibrium will not. Default value is 0. Normal thermal equilbrium models can be calculated with niter=99, although the code seldom requires more than a few iterations (10 or 20 at most) to achieve thermal equilibrium under normal conditions.

5.11. Radius Exponent (radexp)

New in version 2.2 is the ability to have the gas density variable as a power law in radius, i.e. $n = n_0(R/R_0)^{\rm radexp}$, where n_0 and R_0 are the density and radius at the cloud illuminated face. **NB Note that this creates the possibility for calculations which do not end. This because the citeria for a model to end are either that the input column density is reached, or that the electron fraction falls below xeemin. If radexp has a value $\leq$-1 then the column density integral converges only logarithmically at best, and the specified column may never be reached. If radexp $\leq$-2 the local ionization parameter will increase with radius, and so the gas will not recombine and the xeemin criterion will not be met.**

5.12. Number of Continuum bins (ncn2)

New in version 2.2 is the option to control the number of continuum bins. Continuum bins are logarithmically spaced between 0.1 eV and 40 keV, and are calculated according to:

$$\Delta\varepsilon/\varepsilon = (40\mathrm{keV}/0.1\mathrm{eV})^{1/(0.49\mathrm{ncn2})}$$

ncn2 must be in the range between 999 and 99999. The higher value is appropriate for use in modeling X-ray grating spectra. The lower value is appropriate for models where only integrated line luminosities or ionization fractions are desired. Execution time scales approximately proportionately to ncn2.

5.13. Loop Control (loopcontrol)

Used by XSTAR2XSPEC to track each model generated. This value should never be manipulated by the end user.

5.14. mode

Used by the XPI interface. This value should never be manipulated by the end user

Chapter 5

XSTAR output

The primary format of output data for XSTAR is FITS. The files generated by a typical XSTAR run are described below.

Users of version 1 will recognize that the freedom to choose the output format has been eliminated in the interests of simplicity. The most important physical quantities, such as line and continuum luminosities, ion abundances, and temperature are output automatically to fits files. In addition, a concise summary of the radial temperature and ionization structure is output to both the screen and a log file. The user has the freedom to select an additional detailed fits output of level populations by setting an input switch.

1. The Spectral Data File: xout_spect1.fits

The continuum luminosities and optical depths are printed in columns to this ascii fits file. For each energy channel we print: channel index, energy (eV), transmitted and reflected luminosity (in units of 10^{38} erg s^{-1} erg^{-1}) with lines binned and added to the continuum. New in version 2.2 is the inclusion of a column which is just the flux scattered in resonant lines. The final two columns are optical depth in the forward and backward directions (see Chapter 9 for a more detailed description of the meaning of these quantities).

2. The Continuum File: xout_cont1.fits

The continuum luminosities and optical depths are printed in columns to this ascii fits file. For each energy channel we print: channel index, energy (eV), transmitted and reflected luminosity (in units of 10^{38} erg s^{-1} erg^{-1}), and optical depth in the forward and backward directions. In this file lines are not added to the continuum (new in version 2.1).

3. The Line Lumnosity File: xout_lines1.fits

The luminosities and optical depths of the 500 strongest emission lines are printed in columns to this ascii fits file. For each line we print: line index, wavelength ($\mathring{A}$), ion, lower level, upper level, reflected and transmitted luminosity (in units of 10^{38} erg s^{-1}), and optical depth in the forward and backward directions (new in version 2.1).

4. The Abundances Data File: xout_abund1.fits

Print ion abundances and heating and cooling rates in an ascii fits file. For each ion with fractional abundance (relative to its parent element) greater than 10^{-10} the following information is printed: ion index, ion name, fractional abundance (relative to the relevant elemental abundance), abundance (relative to the total hydrogen abundance), and that ion's contributions to heating and cooling rates (in erg cm^{-3} s^{-1}). The elements are ordered by increasing nuclear charge, ions by increasing free charge. Also printed are the Compton and total heating rates, and bremsstrahlung, Compton, and total cooling rates (in erg cm^{-3} s^{-1}). This information is saved at every spatial zone and printed out when the model is complete. A second extension onto this file contains the column densities of the ions at the completion of the model.

5. Detailed Ionic Information: xout_detail.fits

Print all level populations and continuum emissivities for all spatial zones to an ascii fits file. This file is large and time-consuming to view and manipulate, and is only produced if the hidden parameter write_switch is set to 1.

6. Detailed Line Information: xout_detal2.fits

Print all line emissivities for all spatial zones to an ascii fits file. This file is large and time-consuming to view and manipulate, and is only produced if the hidden parameter write_switch is set to 1 (new in version 2.1).

6.1. XSTAR Run Log: xout_step.log

Print input parameters, and a log of the temperature and other useful quantities (radius, $\Delta R/R$ the fractional distance from the illuminated cloud face, column density, ionization parameter, electron fraction, proton number density, temperature, fractional heating-cooling rates, continuum optical depth at the Lyman continuum in the transmitted and reflected directions, and the number of iterations required to reach thermal equilibrium. This is the same as the information printed to the screen. In addition, at the end of a model calculation the luminosities of the 1000 strongest lines are printed, sorted by luminosity, along with the energy budget: total energy absorbed, emitted in the continuum, emitted in lines, and the fractional difference between the first quantity and the sum of the latter two. Models with energy budget errors greater than a few percent should likely be rerun with smaller value of emult.

Chapter 6

Problems and Pitfalls

XSTAR has been designed to be as 'user-friendly' as possible while still maintaining a large amount of flexibility. However, experience has shown that it is difficult to guard against the many possible misuses of the code, and that it is impossible to generate a code which is completely free from errors or unintended features. In this chapter we list what we consider to be some of the most probable pitfalls of the use of XSTAR. This chapter should be read by any user who runs XSTAR2XSPEC or who ventures very far beyond the default parameter values of XSTAR itself.

0.2. Nickel and Iron-peak Eelements

The atomic rates for nickel are less accurate, and less well debugged, than those for other elements. It is recommended that any model be run first with the nickel abundance set to zero. Substantial differences between models with zero and non-zero nickel, in terms of temperature, opacity, etc., should be treated with great caution.

Similar comments apply to elements introduced in version 2.2: Li, Be, B, odd-Z elements between F and K, plus the iron-peak elements (except for Fe itself). Many of the rates for these elements are scaled hydrogenic, most ions with 3 or more electrons have a level structure which is assumed hydrogenic. Quantitative results affected by these elements should be regarded as reliable only for highly ionized models, which are dominated by H- or He-like ions.

0.3. Mtables

If you are using mtables with variable abundances, then you will likely get totally unphysical results unless you read the section 'Important Notes on Mtables' in the xstar2xspec chapter.

1. Execution Time

XSTAR is designed to strike a balance between accuracy and speed, but this inevitably involves some disparity between different computing platforms. As a result, many problems of interest require large amounts of time on machines which are relatively slow, or which are heavily used for other tasks.

In an effort to avoid wasted time (and CPU cycles) we offer the following suggestions: (i) XSTAR does not attempt to calculate ionization, excitation, etc. for elements whose abundances are specified to be less than 10^{-15} relative to hydrogen. Large reductions in computation time can be achieved by zeroing the abundances of elements which are likely to be unimportant anyway: calcium, argon, and nickel. (ii) For some purposes constant temperature is an adequate approximation, and is often a useful preliminary step in deciding parameter values such as column density, and require a fraction of the execution time of full thermal equilibrium models. (iii) For some purposes a low column density ($\leq 10^{18}$ cm^{-2}) will provide sufficient information. Large column densities require significantly more execution time. If large columns are needed, then execution can be speeded up by use of a large value of emult, or a small value of taumax, ans by setting npass to 1. (iv) New in version 2.2 is the ability to specify fewer continuum bins on input. Execution time is approximately proportional to the number of continuum bins.

The parameter files included in the source tree for both xstar and xstar2xspec are set to perform constant temperature models, in order to allow the user to become familiar with xstar without requiring large investments in computer time.

2. Low Density

Although all rates should extrapolate correctly at low density, the level population calculation requires the inversion of large matrices of rates. In most cases, the largest elements of the matrix are the spontaneous decay rates (A values) for allowed transitions. For highly charged ions these can exceed 10^{13} s^{-1}. At low densities the smallest rates of interest are the collisional rates and recombination rates. XSTAR attempts to avoid inverting singular matrices by discarding rows and columns whose largest elements differ from the largest element in the matrix by more than the machine precision (and then assuming the populations in the associated levels are zero). This can have the effect of producing inaccurate solutions particularly at low densities since some physically important transitions (notably recombination) may be discarded. Unfortunately, there is no clear way of automatically informing the user when this is happening. A rough rule of thumb is that densities less than approximately 1000 should be avoided when iron may be ionized beyond Fe XVII.

An indication of possible numerical problems is given by the final integer on each line of the output log file. This is the number of iterations required in order to reach thermal equilibrium and charge neutrality. Models with good convergence will typically have values of 5 or less for this quantity, except for the first step and possibly near ionization fronts. Otherwise, if this integer is large (i.e. greater than, say, 20), and if the value of heating - cooling ('h-c') is greater than 1 – 2%, then it is possible that the density is lower than can be treated accurately by XSTAR.

3. High Density

All level populations are affected by collision rates to and from the superlevels. These rates are calculated by fitting to more complete population kinetic calculations involving hundreds of levels; these fits are valid only up to (electron) densities of 10^{18} cm^{-3}. Attempts to use densities greater than this will result in the code stopping with a message.

4. The Energy Grid

Many of the most important components of the XSTAR calculation require numerical quadratures over energy, and these are generally carried out using straight-forward trapezoid quadratures over a fixed grid of energies. In addition, the computation is speeded by the use of a strictly logarithmic grid spacing in energy. We use 9999 energies spaced logarithmically from 0.1 eV to 20keV. This results in a 0.12% grid spacing, corresponding to, e.g. 8.6 eV at 7 keV. This is the energy resolution of the code.

5. The Ionizing Spectrum

The ionizing spectrum has the obvious effect of creating ionization in the gas. But it also can influence the heating and cooling via Compton scattering if the gas is highly ionized. The standard ionizing spectrum options apply to all the energies in the grid. Therefore if, for example, an ε^{-1} power law is chosen the temperature in Compton equilibrium will be kT=$(\varepsilon_{max} - \varepsilon_{min})/(4\ln(\varepsilon_{max}/\varepsilon_{min}))$, and power law indeces which are greater (or less) than 1 will be influenced even more strongly by the choice of minimum (or maximum) energy. It is likely that the choice we have made in designing the code will not be the choice which is physically appropriate for the situation of interest, so the user is encouraged to input the spectral model from a file, with the appropriate cutoffs built in, in situations where power law spectra and Compton heating/cooling are important.

6. The use of critf

This input parameter allows the user to control the size of the matrix solved when calculating level populations.

In versions 2.1lxx and before this matrix had a maximum size of 2400, and an attempt to solve for more than this number of levels simultaneously would result in xstar stopping with a message 'ipmat too large'. This is most likely to occur in situations where more than 4 or 5 of the lower ion stages of iron meet the critf criterion; for problems with zero iron abundance critf can probably be specified as small as 1.e-15; when iron is non-zero the limit may be reached if critf is less than 1.e-8.

With version 2.2 this is no longer true and small values of this parameter are recommended.

7. Energy Budget

In models where thermal equilibrium is imposed the total amount of energy absorbed from the incident radiation field should balance the total emitted energy in lines and continua. This constraint is not automatically satisfied, since the algorithm for calculating heating and cooling rates is based on the assumption that each spatial zone is at most marginally optically thick. As a check the total energy absorbed and emitted from the radiation field is printed at the end of the log file, along with the fractional error in energy conservation. An error greater than a few percent indicates an inaccuracy in the model results which may significantly compromise emission line strengths, for example. Such models should propbably be rerun with smaller values of emult.

8. Column Density

Xstar chooses spatial steps using a Courant condition (with limits) based on the local opacity. It doesn't pay attention to where the cloud boundary is during this procedure, but rather checks after each step to see if it has gone too far. If yes, it stops. It does not go back and redo the step if it has gone significantly past the column density limit specified in the parameter file. If this problem appears, you need to change the limits of the step size calculator to something smaller. This is done using the parameter emult (normally hidden) in xstar.par. The default is 0.5; values smaller by a factor 5 – 10 may slow execution somewhat but will solve the problem.

9. Notes regarding equivalent widths of unresolved absorption lines in mtables

The mtable results are qualitatively different between version 2.1h and 2.1j and later for the depths of lines when the value of vturbi is small. These differences are entirely due to the challenge of modeling absorption in a binned spectrum. Since before v2.1h the ionization and thermal balance has changed very little.

Taking O VIII L alpha as an example. A cloud with colum $10^{21.3}$, $\log(\xi)$=1.8 and a γ=2 power law ionizing spectrum has a line center optical depth of 1.6e3 for this line, and a temperature $10^{5.13}$ K. So the thermal Doppler velocity is 11 km/s, or 2.4e-3 eV in energy units. The line wavelength is 18.9689 A, or 653.62 eV. The nearest xstar bin boundaries are at 653.1 and 654.0 eV, which are both many Doppler widths away. These versions of xstar both use a logarithmic energy grid evenly spaced between 0.1 and 20000 eV, with E/DeltaE=722, for a total of 9999 energy bins (there are some extras tacked on at high energy). The version in development has 10 times as many, but still will not resolve the lines such as O VIII L alpha.

How to put an unresolved line on a fixed grid? There are 3 obvious ways, 2 of them dumb: Version 2.1h and before simply calculated exp(-tau_line) at the nearest energy grid point, so O VIII L alpha was black in one bin. This version also did not take into account the damping wings accurately. Version 2.1j and later use an accurate Voigt profile, with damping wings, and evaluate the profile function at the grind point. So the bin at 654 eV in my example, being 12 Doppler widths away, has an optical depth of about 0.4. And there was an error in the implementation of this in versions 2.1j and 2.1k when turbulent broadening is included, but that is now fixed.

The third way would be give the bin a depth which would give the correct equivalent width for this unresolved line, since neither of the previous two will do that. This is work, since it requires implementing a curve of growth calculator which can handle damping, and arbitrary values for hte line wavelength relative to the bin boundaries, and can smoothly go over to something like what is in 2.1k when the line is resolved. It is planned to put this in the next version of xstar.

9.1. Bug in version 2.1k

The released version 2.1k has been found to have a bug in the Voigt profile calculation of absorption line profiles. This causes Infs in the opacity for lines with small damping parameters on some machines. Xstar itself does not halt due to this error, and produces optical depths which are large in the core of the affected lines. Xstar2xspec can halt with an arithmetic error caused by the inability of fitsio to read the large opacity. This bug has been repaired in version 2.1kn and later.

Chapter 7

XSTAR2XSPEC

To facilitate using XSTAR with actual data, XSTAR2XSPEC was developed. XSTAR2XSPEC is a perl script which calls XSTAR multiple times and generates table models from the results of these simulations which can then be utilized for model fitting in the XSPEC spectral fitting program. XSTAR2XSPEC also generates a log file which is a concatenation of all the xout_step.log files from all the XSTAR runs. This is a convenient way of generating a grid of models for other purposes, such as studying the dependence of line strengths on various input parameters.

1. Parameters

The parameter handling for XSTAR2XSPEC is designed to be as flexible as possible, in principle, limited only by the physical resources (RAM, disk space & CPU time) available on your machine. You have the choice of varying *any* of the physical parameters used as input in an XSTAR model.

1.1. Physical Parameters

XSTAR models are based on 21 physical parameters described in detail in Chapter 4. In quick summary, they are cfrac, temperature, pressure, density, trad, rlrad38, column, rlogxi, habund, heabund, cabund, nabund, oabund, neabund, mgabund, siabund, sabund, arabund, caabund, feabund & niabund. Each of these parameters needs at least one additional parameters and as many as four additional parameters to specify is variation during the program run.

Each parameter has three levels for classifying its variability:

Constant (variation type = 0): This parameter is held constant in all the XSTAR runs.

Additive (variation type = 1): The Additive class of parameters provides a simple method for varying parameters that are reasonably independent of the others. For more info on how additive parameters function, see the XSPEC manual and OGIP Memo OGIP 92-009.

Interpolated (variation type = 2): Interpolated parameters provide the greatest accuracy in building table models. They also require the most processing time. For each interpolated parameter, you can define some number of points between a maximum and minimum range. The placement of these intermediate points is determined by the interpolation type – linear (interpolation type = 0) or logarithmic (interpolation type = 1).

In estimating the running time of XSTAR2XPEC, the key factor is the number of times XSTAR is called. Consider a run with $\mathrm{N_I}$ interpolated parameters, where interpolated parameter i is evaluated at n_i points ($1 \leq i \leq \mathrm{N_I}$). The total number of times XSTAR is called is then $\prod_{\mathrm{i=1}}^{\mathrm{N_I}} n_i$. However, if $\mathrm{N_A}$ additive parameters are also defined, then for each set of interpolated variables, there is one run with all the additive parameters are zero and the remaining $\mathrm{N_A}$ runs have one of the additive parameters at it's maximum value and the rest all zero. This means that the total number of times XSTAR must be run is given by

$$(\mathrm{N_A}+1)\prod_{\mathrm{i=1}}^{\mathrm{N_I}} n_i \tag{7.1}$$

and this can provide you with a feel for how long a complete XSTAR2XSPEC run will require. As an example, if we defined 2 interpolated parameters (one evaluated at 5 points and the other evaluated at 4 points) and 8 additive parameters, XSTAR would be run a total of

$$(8+1)(5)(4) = 180 calls \tag{7.2}$$

which means that if the XSTAR runs for the appropriate model range averages five minutes, it will take approximately (180)(5 minutes) = 900 minutes = 15 hours for a complete XSTAR2XSPEC run.

1.2. XSTAR Fixed Parameters

These values are the same for *all* XSTAR runs.

1.3. XSTAR2XSPEC Control parameters

elow: This parameter determines the low energy end (in eV) of the spectrum selected from the XSTAR output files.

ehigh: This parameter determines the high energy end (in eV) of the spectrum selected from the XSTAR output files.

2. Running XSTAR2XSPEC

To run XSTAR2XSPEC, simply type

```
xstar2xspec [options]
```

at the system prompt, where [options] consists of one or more of the options listed below. Generally, you will run XSTAR2XSPEC with no options. There are a couple of control options which are described below. Currently the script prompts you for *all* parameters by invoking the pset utility (included in the standard XSTAR package). The defaults are adequate for most situations.

Entering all the many parameters prompted by XSTAR2XSPEC can be tedious, and prone to error. An alternative is to edit the parameter file using an FTOOL routine or a text editor. Since XSTAR2XSPEC is actually a perl script and not an FTOOL, it does not have its own parameter file. Rather, it calls several other FTOOLS which do have their own parameter files. The FTOOL responsible for setting up the multiple commands to call XSTAR is called XSTINITABLE, and all the relevant parameters can be set by editing the parameter file xstinitable.par. This is what we do when using this script.

XSTAR2XSPEC Options:

`-save:` Save the spectral FITS files, modifying the file name to include the value of the loopcontrol variable for better identification. Note this can use GBs of disk space.

`-verbose:` Generate more diagnostic messages (applies to the XSTAR2XSPEC script only).

`-restart:` Continues the XSTAR2XSPEC run using the previous run. Note that it does *NOT* check the integrity of the table files from the terminated run. This is the user's responsibility.

2.1. Output

Four files are produced by xstar2xspec:

1. xstar2xspec.log is a concatenation of all the xout_step.log files from all the xstar runs called by xstar2xspec.
2. xout_ain.fits is a fits file containing the atable with the reflected emission spectrum produced by xstar. This file is ready for use by xspec using the 'model atable xout_ain.fits ...' command.
3. xout_aout.fits is fits file containing the atable with the emission spectrum in the forward (transmitted) direction produced by xstar. This file is ready for use by xspec using the 'model atable xout_aout.fits ...' command.
4. xout_mtable.fits is a fits file containing the mtable with the absorption spectrum in the forward (transmitted) direction produced by xstar. This file is ready for use by xspec using the 'model mtable xout_mtable.fits pow ...' command.

2.2. Important Notes on Mtables

If you are using mtables with variable abundances, then you will likely get totally unphysical results unless you note the following.

The mtable models are constructed using two kinds of parameters: interpolated parameters and additive parameters. Interpolated parameters are treated as one might expect: if the model is represented by a vector $M_i(x_j)$, corresponding to the model flux at various energies ε_i and stored at various values of the free (intepolated) parameter x_j, then for a value of the parameter not on the tabulated grid xspec calculates the model value as

$$M_i(x) = \Sigma_j M_i(x_j)\omega_j$$

where ω_j are suitably chosen weights for, eg. linear interpolation. Ths is in contrast to the treatment of additive parameters: for an mtable, xspec needs the value of the model tabulated at only two values of the free additive parameter y, 0 and y_{max}. Then the model is calculated for some arbitrary y as

$$M_i(y) = (M_i(y_{max}) - M_i(0))\frac{y}{y_{max}} + M_i(0)$$

i.e. it is assumed that the model scales linearly with the value of the additive parameter y. This formalism was developed for emission models, where emissivities might be expected to scale linearly with elemental abundance. In the case of absorption models, the value of M which xspec uses is a transmission coefficient, and this does not scale linearly with abundance. Rather, the transmission coefficient is related to the optical depth by:

$$M_i(y) = e^{-\tau_i(y)}$$

and the optical depth $\tau_i(y)$ does scale (approximately) linearly with abundance. For this reason, xspec has incorporated the etable models, in which the model builder supplies optical depths rather than transmission coefficient, linear interpolation is used to calculate the optical depth for a given abundance, and then the transmission coefficient is calculated using the above expression. The results:

(1) If you use an mtable calculated by xstar2xspec using varable abundances (i.e. any additive parameters allowed to vary) in xspec using the 'model mtablexout_mtable.fits' command, you will get unphysical results. It is easy to see that if $M_i(0)$ is non-zero (which it often is owing to 0 abundances) then you will never get deep absorption.

(2) If so, you should use the 'model etablexout_mtable2.fits' command in xspec instead. Entering parameter, etc., in xspec is the same for etables as for mtables.

(3) BUT, the etable itself is supposed to be filled with optical depths, not transmission coefficients. The tables made by xstar2xspec are transmission coefficients, and they need to be converted. This is a simple transformation: replace the contents of every cell of the model extension of the mtable file (x_{ij}) with -ln(x_{ij}). This can be done various ways. One way is to use the calculator inside of the ftools gui routine xv, one column at a time. This has been done for the various precalculated models on the website, and these are stored with the names xout_mtable2.fits.

Some Specific Examples, and a Discussion of the Analytic Model warmabs

When fitting to absorption spectra it is important to be aware of the inherent limitations of xstar when applied to the fitting of absorption. There are two distinct reasons for this: spectral resolution effects, and interpolation effects. In order to illustrate these effects, we reproduce here a discussion between TK and P. Oneill of Imperial College in which these questions are raised, and some of the issues are discussed.

Question:

I've been using some of the sample XSPEC models from the current version of XSTAR to model a warm absorber.

I've been using the grid18 models with the abundances fixed at their default values. I found that using the XSPEC model "mtablexout_mtable.fits*powerlaw" gave different results (less absorption) than when I used "etablexout_mtable_2.fits*powerlaw". I thought that these should provide the same results, so long as the abundances are not changed. Am I using these tables correctly, or is there perhaps a problem somewhere?

Answer:

The problem is that variable abundances, they are treated as additive parameters in xspec tables, really can't be treated accurately using either type of table. If you use an mtable, then the transmittivity in a bin, i, at a given ionization parameter and column, is:

$$M_i = M_i^0 + \Sigma_j x_j M_i^j \tag{7.3}$$

where M_i^0 is the 'zero abundance' version of the model at that ionization parameter and column, x_j is the abundance of element j, and M_i^j is the model calculated with that element set to unity (relative to cosmic). The first problem with this approach is that tranmission doesn't scale this way with abundance. If you double the abundance of an element, you should double the optical depth. Doubling the transmittivity has both the wrong qualitative and quantitative behavior. But an even bigger problem with this is that the whole thing, M_i, multiplies your continuum. It is almost guaranteed to bigger than 1 if some of the x_j are non-zero. So this is totally the wrong formulation to use when the abundances vary.

In the case of an etable the transmittivity is:

$$M_i = exp(-(E_i^0 + \Sigma_j x_j E_i^j)) \tag{7.4}$$

where $E_i^0 = -\ln(M_i^0)$ etc.

This is better, since it predicts the correct dependence on abundance, i.e. that the optical depth scales approximately linearly with abundance.

But it is important to remember that what you are trying to simulate is a 'real' photoionized gas with some abundance set x_j that fits your data. What you are using to fit to your data are M_i^0, which in the case of xstar are models calculated with only hydrogen and helium, and the set of M_i^j, which are supposed to be calculated with pure element j.

The first problem is that the cooling, and therefore the temperature, depends on the element abundance in the gas, and this is not linear. The temperature couples back into the ionization balance, and that affects the opacity, transmittivity, etc.

The second problem is that, while it is straightforward to calculate a model with no elements heavier than H and He, it is not straightforward to calculate a model with pure iron, or calcium, say. That's because H and He havs nice simple cooling properties at low temperature, and their opacity is simple, etc. Pure iron models at best are likely to be very different than H+He+Fe models, or from a cosmic mix. So what I do is make the M_i^j using H+He+element j in cosmic ratios. But the problem with this is that then the opacity due to H and He are present *in every* M_i^j*, and* E_i^j, particularly at low ionization parameters. So if you sum over cosmic abundances using an etable you are including the H and He opacity with every one of the E_i^j. At high ionization parameter ($\log(\xi) > 1$?) this should not be a problem because H and He are ionized sufficiently that they do not contribute to the opacity. But at low ionization parameter, the model spectra calculated using etables will have significantly more low energy opacity due to the multiple counting of H and He than they would in a real xstar model with the corresponding parameters.

The strategy to use at low ionization parameters therefore is to use the etable to find a reasonably good fit to the spectrum with xspec, and then run xstar2xspec and make a very small table, i.e. just 1 column and 2 ionization parameters (xspec needs tables to have at least 2 entries), with constant abundances, set to the values which come from the fit. Constant abundance grids (such as grid19) when used as mtables do not suffer from any of these problems.

I attach 4 figures to illustrate these problems. All use the mtable or etable from grid18. All are plotted between 0.6 and 0.7 keV, power law continuum, index=1, norm=1, $\log(\xi)$=1. The continuum level at .6 keV is 1.66.

Comparing these shows the big problem with mtables with variable abundances:

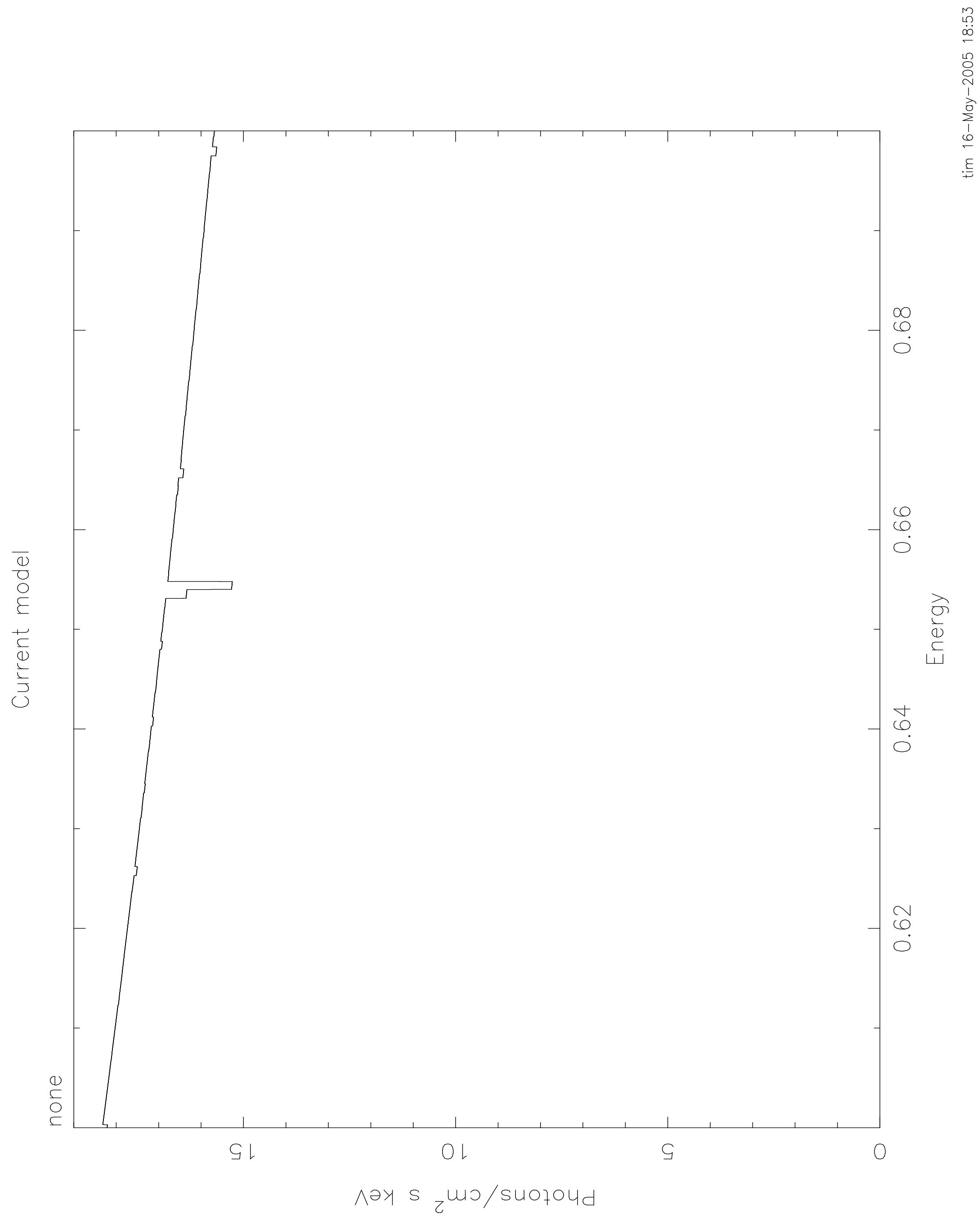

Fig. 1.— figure 1: cosmic abundances, mtable

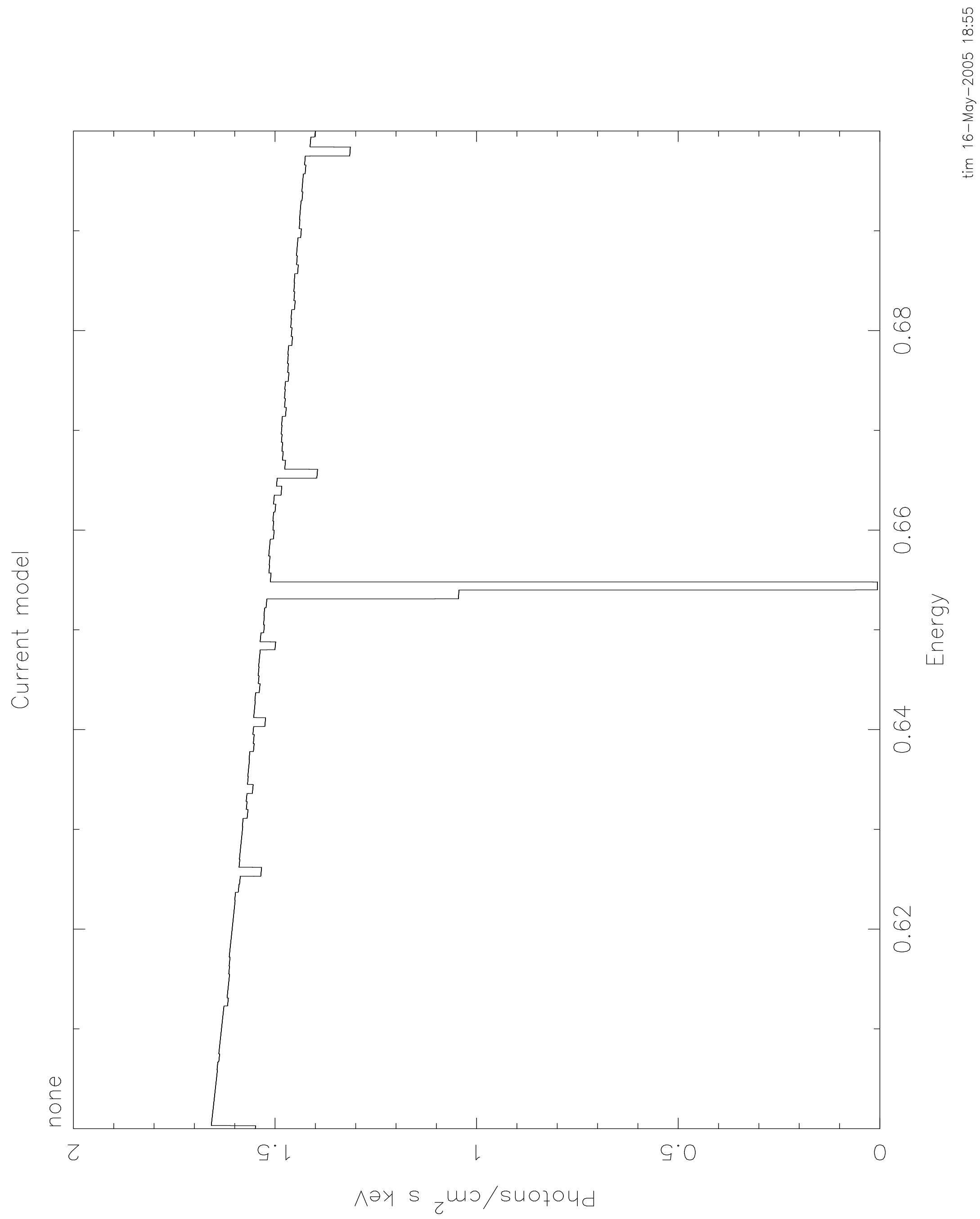

Fig. 2.— figure 2: cosmic abundances, etable

they give a transmittivity which is >1. In this case, the transmittivity is >10.

Here again, the problem is that even the pure oxygen mtable has the transmittivity of the 'zero abundance' model, M_i^0, which really has pure H and He and therefore has unit transmittivity. The line never can go black. The only model which is nearly correct is 3, but here again you have the problem of multiple counting of the H and He opacity. But at this high ionization parameter that is negligible. Anyway, I hope you understand the problem.

It is straightforward to use xstar2xspec to calculate grids without variable abundances. THEN you can use mtables, and there should be none of these problems.

Question:

I have made some comparisons between vturb=0 models and those with vturb=100. I find that vturb=0 gave very different results (much less absorption) to the vturb=100 grid18 etable. I hadn't expected to see the large difference with a change from vturb=100 to vturb=0. Is this expected behaviour?

Answer:

vturb can make a significant difference in the results of xstar modeling of absorption spectra, and the reason is at least partially due to numerics. In calculating the energy dependent opacity Xstar simply evaluates the profile function for the line at each grid energy. So if the line width is less than, or comparable to, the xstar grid spacing, then maximum depth of a given absorption line in the mtable may be very different from the true depth at line center. Obviously this problem is more severe for small vturb, i.e. narrow lines. The total line equivalent width does not depend on vturb, but the numerics do not reflect that unless the line is broader than the grid resolution. And the limiting resolution is the internal xstar grid spacing, which corresponds to 420 km/s, since this is what is used in constructing the table. This is another flaw in the use of tables for fitting to spectra.

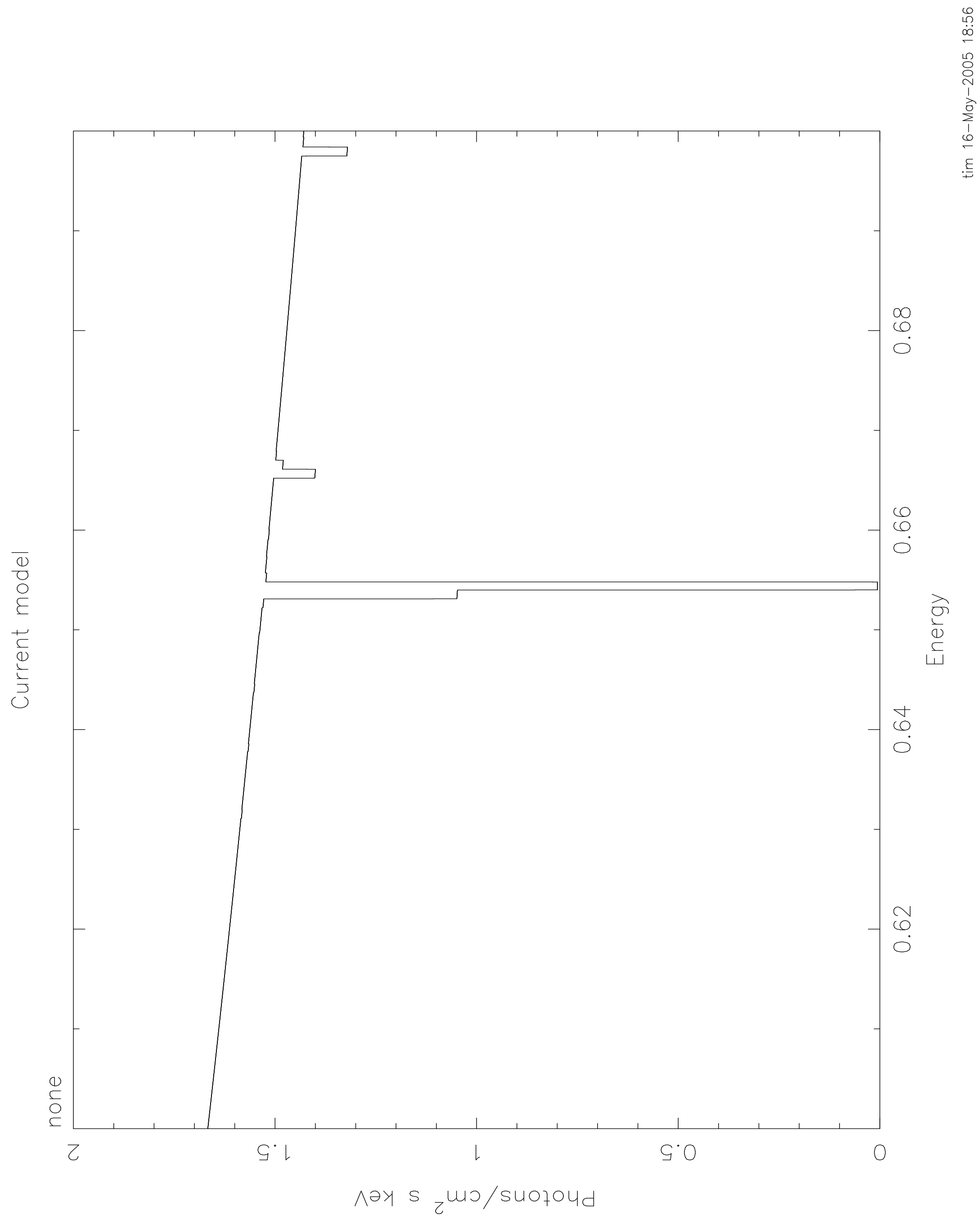

Fig. 3.— figure 3: oxygen=1, other metals=0, etable

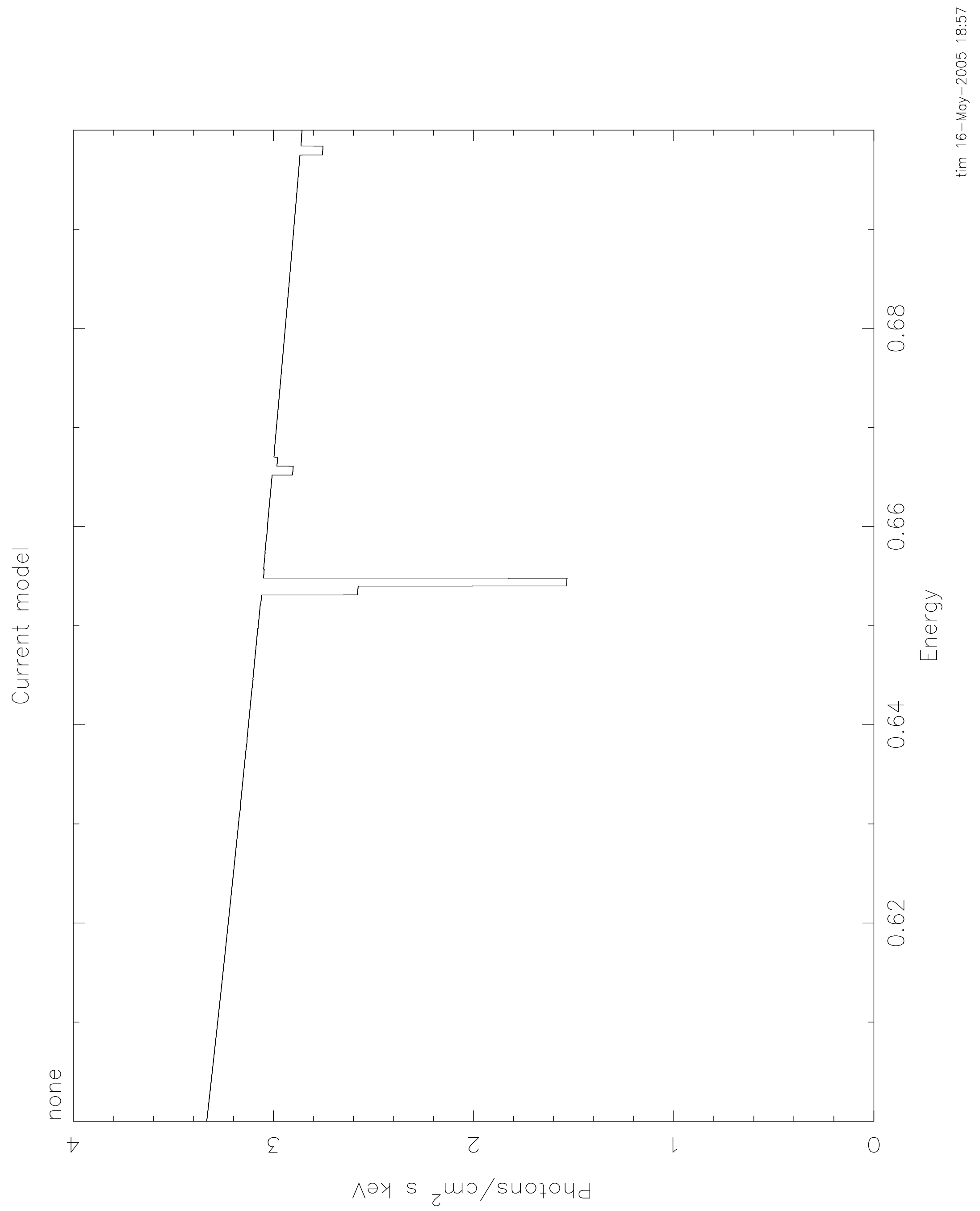

Fig. 4.— figure 4: oxygen=1, other metals=0, mtable

There is now subroutine which calculates xstar warm absorber (called 'warmabs') and warm emitter (called 'photemis') spectra and which can be called as an xspec 'analytic' model. This calculates absorption spectra 'on the fly', and so will evaluate the profile function on whatever grid you use in xspec. So you can choose to resolve all the lines if you want, using the 'dummyrsp' command with a very large number of bins. It is available on the xstar web site, look for the link under 'XSTAR news'. It also does not suffer from the problems discussed previously about the approximations inherent in variable abundance. It calculates the opacity directly from a stored library of level populations calculated for a generic power law grid.

One drawback of this routine is that it brute-force calculates Voigt profiles and opacity for all the lines and bound-free continua which are above a certain threshold in strength, and so it is time-consuming. Currently calls to this routine can take up to 30 seconds on a slow or busy machine. It will be much faster on a faster machine, or if there are few elements with non-zero abundances, or if only a narrow spectral range is of interest, or if the ionization parameter is frozen.

Another drawback to this routine is that the current version assumes constant ionization throughout a slab of given ionization parameter and column. Real slabs will have lower mean ionization in the deeper, shielded regions, and this lower ionization material will absorb more efficiently. This means that the current warmabs assumption results in an underestimate of the opacity, particularly at low energies, for slabs with large column. This will be remedied in future versions of this routine.

Yet another issue which deserves mention is that of interpolation and grid spacing in the interpolated parameters as implemented by xspec. Many of the available xstar grids were calculated with models spaced 1 decade apart in column. This has the potential to lead to inaccuracies in the model spectrum calculated by xspec, since it will interpolate between grid points. Sparsely gridded models may miss important details. Grids 19b and 19c provide an example.

Question:

I have compared the grids with the warmabs analytical model. These are in firgure 5. 19c is the solid line, warmabs is the dashed line, and 19b is the dotted line. The model here is the same model that I used to fit the data using 19c (ie, it has 1.4e20 of wabs also). I hope you can get some info from this. The parameters used for these models were: logξ=1.27, N=5.89e22.

Answer:

OK, so what I see from your plot comparing the models is that:

1) grid19b and grid19c give qualitatively different results near the best fit parameter values in the strength of the absorption near the K edge of oxygen. This seems to me to be clear evidence for the errors introduced by interpolation in a sparsely gridded table in column, as I suggested.

2) grid19c and warmabs agree well at energies above .8 keV, except for the lines. This seems to me to be a comforting check on the validity of warmabs.

3) there is disagreement in the strength of the absorption below 0.8 keV. We decided that the assumption of a constant ionization slab made in warmabs would tend to underestimate the absorption, since the ionization balance would be lower in the shielded parts of a real slab, hence more absorption. This is the sense of the disagreement, and so seems to make sense. Of course, there is still some interpolation involved in the use of grid19c, since your best fit values for N and $/xi$ are not precisely on the grid values. I have compared warmabs and grid19c for the nearest grid values,

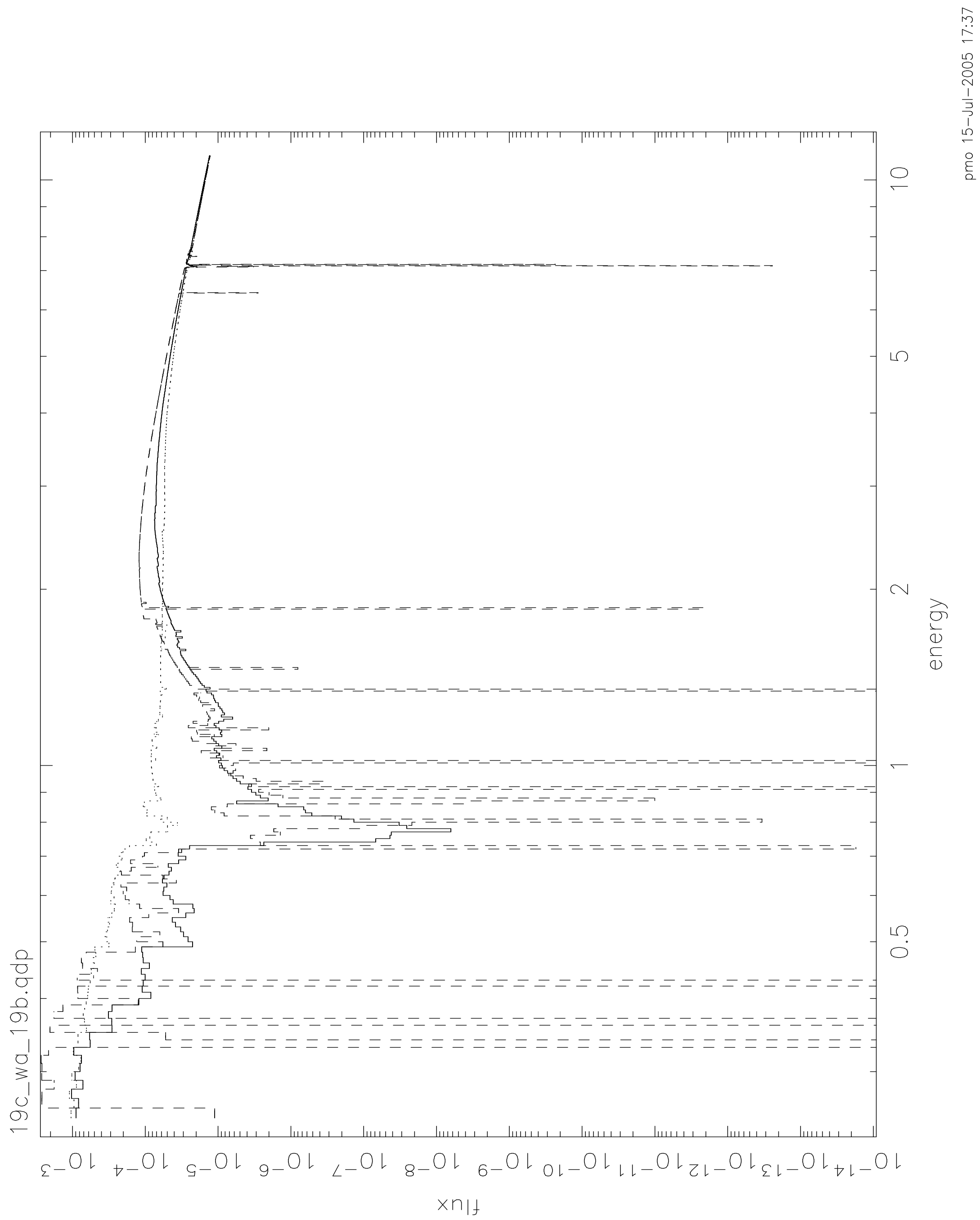

Fig. 5.— figure 5: Comparison of grid19b, 19c, and warmabs

and the effect of interpolation does not appear to change the sense of the disagreement. So this is a strong argument for fixing up warmabs so that it does not make the uniform slab assumption.

4) There is also disagreement between grid19c and warmabs about the strength of the lines. I think this is due to the effect of energy binning: Your energy grid must be rather coarse, i.e.R=E/Delta(E) 100. If you make the same plot with a grid resolution of R=10000, things look better. I attach these plots: figure 6=warmabs, figure 7=grid19c. But there is still disagreement, and this I think is because the grid19c results have the intrinsic xstar internal resolution, which is much less than 10000, and so information is lost about the line profiles, while warmabs calculates the lines specifically for the resolution requested, and so should be more accurate.

2.3. Notes on Normalization

The problem with creating a flexible tool for modelling emission and absorption is that there have several free parameters affecting real spectra, including: source luminosity, distance, reprocessor column density, ionization parameter, and geometry. By geometry we mean covering fraction around the source, which affects emission, and covering fraction across our line of sight to the source, which affects absorption. With the xstar2xspec tables you should be able to model a wide range of choices for this, but there is not a unique one-to-one mapping between the values of these physical parameters and the values used in running xstar and constructing the tables.

The free parameters which can be varied when running xstar2xspec include the abundances, column density, gas density, and ionization parameter. These all have a straightforward intepretation as physical parameters. The emitter normalization and geometry are not uniquely determined, owing to the ambiguity between source luminosity and distance.

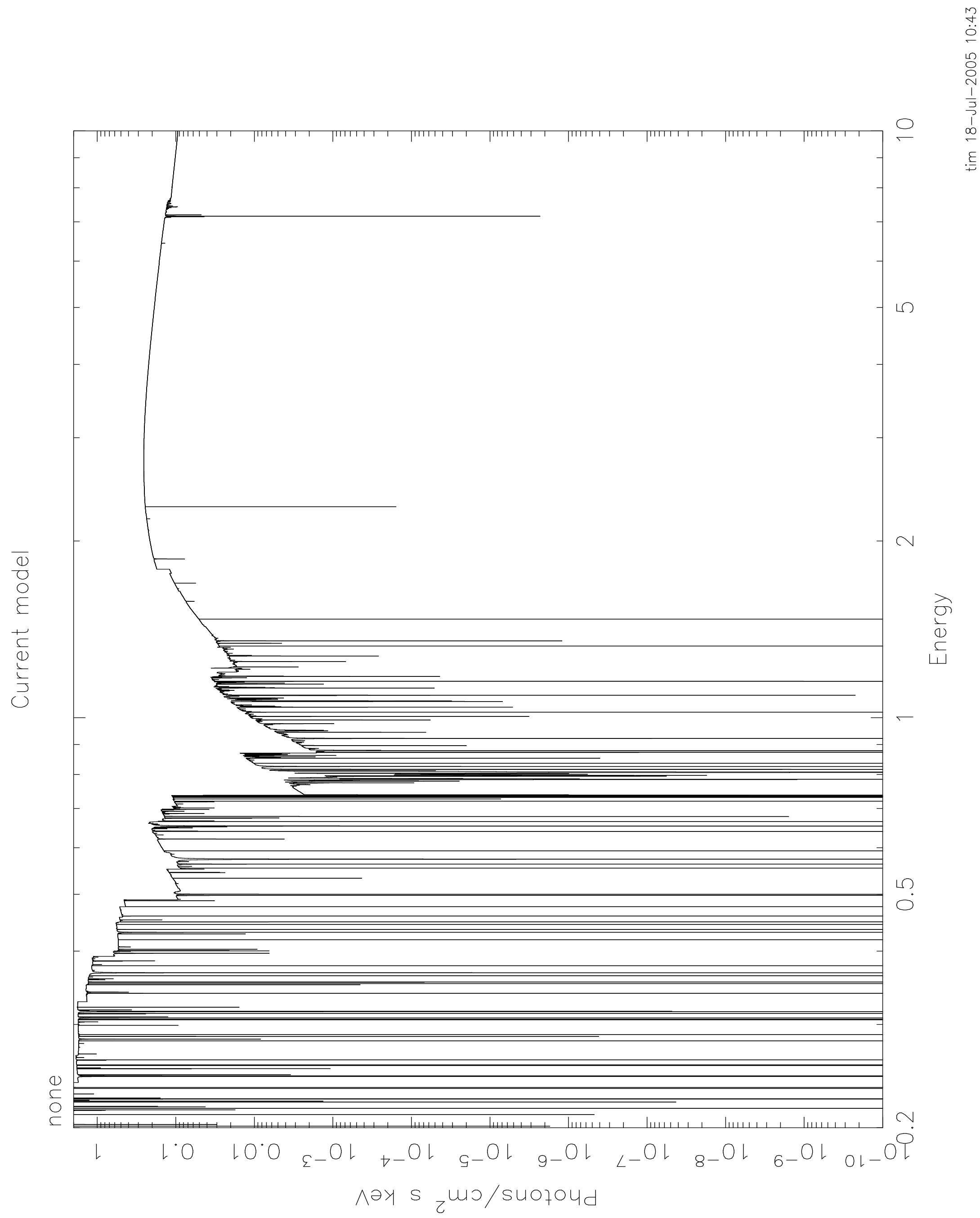

Fig. 7.— figure 6: warmabs model, same parameters as figure 5, R=10000

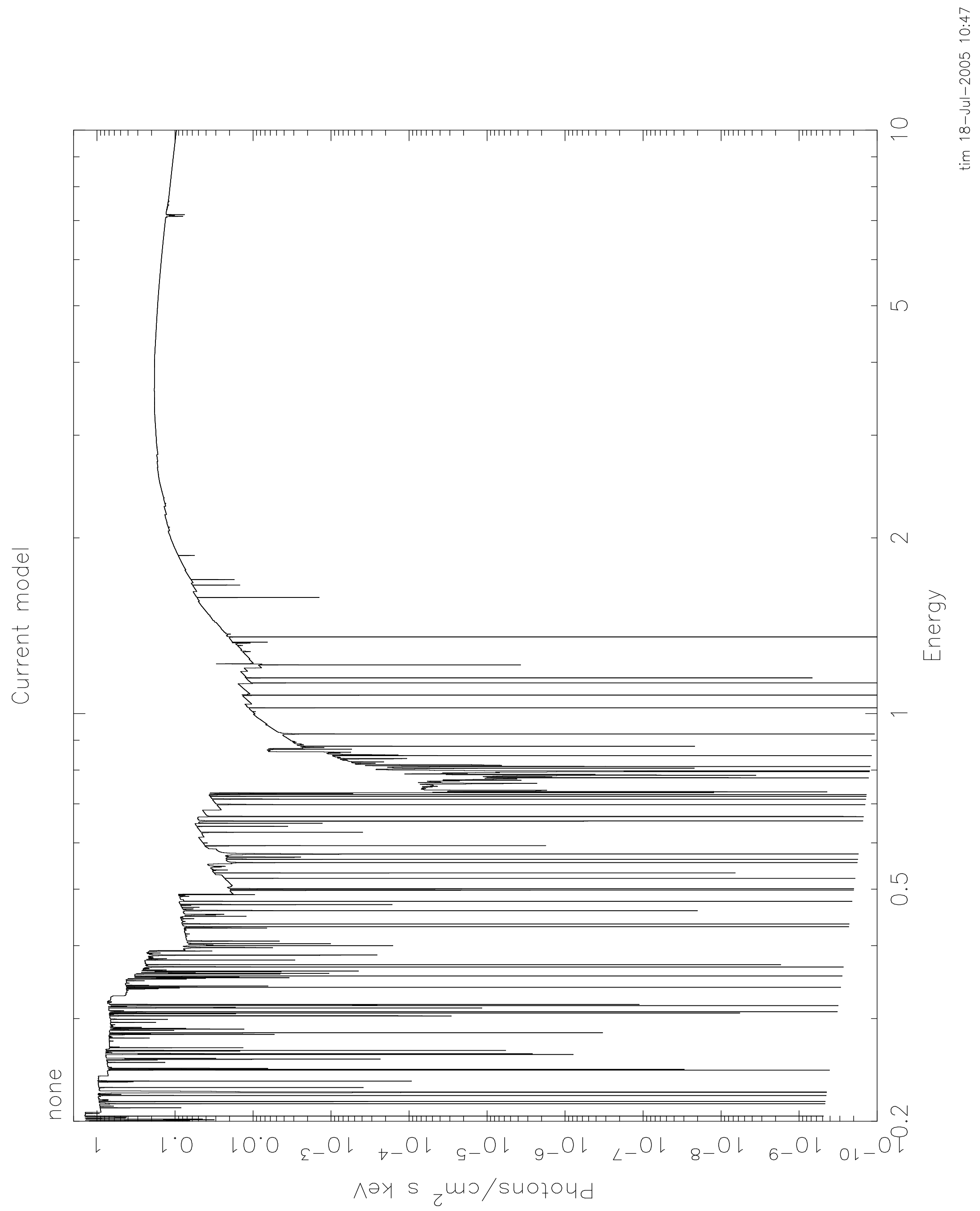

Fig. 8.— figure 7: grid19c, same parameters as figure 5, R=10000

It is helpful here to be very specific, at the risk of being repetitive. The procedure followed by xstar2xspec in making tables is: (i) Generate a sequence of command line calls to xstar (this is done by the tool xstinitable); (ii) step through the calls to xstar; (iii) for each one calculate the appropriate xstar model; (iv) and then take the spectrum output of the model (the file xout_spect1.fits) and convert it to the right units and append it to the fits table (xout_aout.fits, xout_ain.fits, or xout_mtable.fits). The last step (iv) is done by the tool xstar2table. The conversion is as follows: xspec wants a binned spectrum in units model counts/bin for atables. If this is denoted F_n^{mod}, and the luminosity used in calculating the xstar grid is L_{tot}^{xstar} then

$$F_n^{mod} = \frac{L_\varepsilon^{xstar}}{L_{tot}^{xstar}} \frac{10^{38}}{4\pi(1\mathrm{kpc})^2} \left(\frac{\Delta\varepsilon}{\varepsilon}\right)$$

which can be rewritten:

$$F_n^{mod} = \frac{L_\varepsilon^{xstar}}{L_{tot}^{xstar}} \left(\frac{\Delta\varepsilon}{\varepsilon}\right) 8.356 \times 10^{-7}$$

and $\left(\frac{\Delta\varepsilon}{\varepsilon}\right)$ is the fractional energy bin size.

The meaning of this quantity and the normalization can be better understood if we consider how xspec works in more detail. Xspec calculates the model count rates per bin by multiplying the F_n^{mod} vector with the response matrix A_{nm} and multiplying by a normalization factor κ:

$$C_m^{mod} = \kappa \Sigma_n F_n^{mod} A_{nm}$$

The observed count rate C_m^{obs} is calculated from the physical flux recieved by the satellite F_ε^{obs}:

$$C_m^{obs} = \Sigma_n F_\varepsilon^{obs} A_{nm} \left(\frac{\Delta\varepsilon}{\varepsilon}\right)$$

Then it's easy to see that $C_m^{obs} = C_m^{mod}$ if the shape of the model fits the observations and

$$\kappa = \frac{C_m^{obs}}{C_m^{mod}}$$

or

$$\kappa = \frac{F_\varepsilon^{obs}\left(\frac{\Delta\varepsilon}{\varepsilon}\right)}{F_n^{mod}}$$

Now, if the astronomical X-ray source actually resembles the physical scenario assumed by xstar, i.e. if it consists of a shell of photoionized material surrounding a point source of continuum with covering fraction f then $F_\varepsilon^{obs} = fL_\varepsilon^{source}/(4\pi D^2)$, where L_ε^{source} is the actual specific luminosity emitted by the shell, and D is the distance to the source. Then

$$\kappa = f\frac{L_\varepsilon^{source}/10^{38}}{L_\varepsilon^{xstar}/L_{tot}^{xstar}}\frac{1}{D_{kpc}^2}$$

And, if we have gotten the model exactly right and $L_\varepsilon^{source} = L_\varepsilon^{xstar}$ (which implies that $L_{tot}^{source} = L_{tot}^{xstar}$) then

$$\kappa = f\frac{L_{tot}^{xstar}/10^{38}}{D_{kpc}^2}$$

If, on the other hand, the shape of the emitted spectrum is right but the luminosity of the astrophyscial source is different from the luminosity used in calculating the xstar model then $L_\varepsilon^{source} = L_\varepsilon^{xstar}L_{tot}^{source}/L_{tot}^{xstar}$ and

$$\kappa = f\frac{L_{tot}^{source}/10^{38}}{D_{kpc}^2}$$

which can be inverted to find the things on the right hand side if you have a fitted value for κ.

So, for example, if your atable model fits to the data with a normalization=1, let's say, and you used luminosity=1 in creating the table, then this would imply that your data was consistent with a full shell illuminated by a luminosity of 10^{38} erg/s at a distance of 1 kpc, or it also could be a shell illuminated by a luminosity of 10^{44} erg/s at a distance of 1000 kpc.

For another example, let's say you know the distance to the source is 1 kpc and the luminosity is 10^{38}, but the best fit has an emitter normalization of 0.1. This would suggest (to me) that rather than a full sphere, the emitter only subtends 10% of the solid angle around the source.

Obviously, the column density of the emitter is important also. If you don't know the column density, then a shell of column density 10^{19} cm^{-2} illuminated by a 10^{38} erg s^{-1} source will probably have very similar emitted X-ray spectrum to a shell of column density 10^{20} illuminated by a 10^{37} erg s^{-1} source. If there are absorption features in the spectrum, then they may constrain the column density.

2.4. Speeding Things Up

If your disk or CPU resources are limited, you might want methods to reduce the execution time of and XSTAR run. Here are some methods:

1. In most cases, you are interested in the physical conditions in a plasma of fixed composition (usually solar). In this case, you can define the variation type of the composition parameters as zero. This will keep the abundances constant and can reduce the running time by about a factor of twelve.

2. Keep the number of interpolated parameters at a *minimum*. Two is usually

sufficient. Two interpolated parameters sampled at five points each requires 25 runs of XSTAR. Adding two more parameters at the same sampling requires 625 XSTAR runs. With our sample five minute (optimistically) XSTAR run, that comes out to over 52 *hours*!

3. Examples

In what follows we give a couple of examples of the use of XSTAR2XSPEC. Here we provide the entire parameter file XSTINITABLE.PAR. If the desired application resembles one of these, then the user can edit these files and copy them into the pfiles directory.

3.1. Example 1: A grid of coronal models

In this example density is held constant, column density is low, thermal equilibrium is not satisfied, and temperature and element abundances are varied in the manner familiar from models such as APEC or MEKAL.

```
cfrac,r,a,1.,0.,1.,"covering fraction soft maximum"
cfractyp,i,h,0,0,2,"covering fraction variation type"
cfracint,i,a,1,0,1,"covering fraction interpolation type"
cfracsof,r,a,0.,0.,1.,"covering fraction soft minimum"
cfracnst,i,a,1,1,20,"covering fraction number of steps"
temperature,r,h,1000.,0.,1.E4,"temperature soft maximum (/10**4K)"
temperaturetyp,i,h,2,0,2,"temperature variation type"
temperatureint,i,a,1,0,1,"temperature interpolation type"
temperaturesof,r,a,1.,0.,1.,"temperature soft minimum"
temperaturenst,i,a,7,1,20,"temperature number of steps"
```

```
pressure,r,h,0.03,0.,1.,"pressure soft maximum (dyne/cm**2)"

pressuretyp,i,h,0,0,2,"pressure variation type"

pressureint,i,a,1,0,1,"pressure interpolation type"

pressuresof,r,a,0.,0.,1.,"pressure soft minimum"

pressurenst,i,a,1,1,20,"pressure number of steps"

density,r,a,1.E+8,0.,1.E18,"density soft maximum (cm**-3)"

densitytyp,i,h,0,0,2,"density variation type"

densityint,i,a,0,0,1,"density interpolation type"

densitysof,r,a,1.e+10,0.,1.e+18,"density soft minimum"

densitynst,i,a,2,1,20,"density number of steps"

trad,r,a,-1.,,,"radiation temperature or alpha soft maximum?"

tradtyp,i,h,0,0,2,"radiation temperature variation type"

tradint,i,a,1,0,1,"radiation temperature interpolation type"

tradsof,r,a,0.,0.,1.,"radiation temperature soft minimum"

tradnst,i,a,1,1,20,"radiation temperature number of steps"

rlrad38,r,a,1.,0.,1.E10,"luminosity soft maximum (/10**38 erg/s)"

rlrad38typ,i,h,0,0,2,"luminosity variation type"

rlrad38int,i,a,1,0,1,"luminosity interpolation type"

rlrad38sof,r,a,0.,0.,1.,"luminosity soft minimum"

rlrad38nst,i,a,1,1,20,"luminosity number of steps"

column,r,a,1.E17,0.,1.E25,"column density soft maximum (cm**-2)"

columntyp,i,h,0,0,2,"column density variation type"

columnint,i,a,1,0,1,"column density interpolation type"

columnsof,r,a,1.E22,1.,1.E25,"column density soft minimum"

columnnst,i,a,1,1,20,"column density number of steps"

rlogxi,r,a,4.0,-10.,+10.,"log(ionization parameter) soft maximum (erg cm/s)"

rlogxityp,i,h,2,0,2,"log(ionization parameter) variation type"

rlogxiint,i,a,0,0,1,"log(ionization parameter) interpolation type"
```

```
rlogxisof,r,a,1.,-10.0,+10.0,"log(ionization parameter) soft minimum"
rlogxinst,i,a,4,1,20,"log(ionization parameter) number of steps"
habund,r,h,1.,0.,100.,"hydrogen abundance soft maximum"
habundtyp,i,h,0,0,2,"hydrogen abundance variation type"
habundint,i,a,1,0,1,"hydrogen abundance interpolation type"
habundsof,r,a,0.,0.,1.,"hydrogen abundance soft minimum"
habundnst,i,a,1,1,20,"hydrogen abundance number of steps"
heabund,r,h,1.,0.,100.,"helium abundance soft maximum"
heabundtyp,i,h,0,0,2,"helium abundance variation type"
heabundint,i,a,1,0,1,"helium abundance interpolation type"
heabundsof,r,a,0.,0.,1.,"helium abundance soft minimum"
heabundnst,i,a,1,1,20,"helium abundance number of steps"
cabund,r,h,1.,0.,100.,"carbon abundance soft maximum"
cabundtyp,i,h,1,0,2,"carbon abundance variation type"
cabundint,i,a,1,0,1,"carbon abundance interpolation type"
cabundsof,r,a,0.,0.,1.,"carbon abundance soft minimum"
cabundnst,i,a,1,1,20,"carbon abundance number of steps"
nabund,r,h,1.,0.,100.,"nitrogen abundance soft maximum"
nabundtyp,i,h,1,0,2,"nitrogen abundance variation type"
nabundint,i,a,1,0,1,"nitrogen abundance interpolation type"
nabundsof,r,a,0.,0.,1.,"nitrogen abundance soft minimum"
nabundnst,i,a,1,1,20,"nitrogen abundance number of steps"
oabund,r,h,1.,0.,100.,"oxygen abundance soft maximum"
oabundtyp,i,h,1,0,2,"oxygen abundance variation type"
oabundint,i,a,1,0,1,"oxygen abundance interpolation type"
oabundsof,r,a,0.,0.,1.,"oxygen abundance soft minimum"
oabundnst,i,a,1,1,20,"oxygen abundance number of steps"
neabund,r,h,1.,0.,100.,"neon abundance soft maximum"
```

```
neabundtyp,i,h,0,0,2,"neon abundance variation type"
neabundint,i,a,1,0,1,"neon abundance interpolation type"
neabundsof,r,a,0.,0.,1.,"neon abundance soft minimum"
neabundnst,i,a,1,1,20,"neon abundance number of steps"
mgabund,r,h,1.,0.,100.,"magnesium abundance soft maximum"
mgabundtyp,i,h,1,0,2,"magnesium abundance variation type"
mgabundint,i,a,1,0,1,"magnesium abundance interpolation type"
mgabundsof,r,a,0.,0.,1.,"magnesium abundance soft minimum"
mgabundnst,i,a,1,1,20,"magnesium abundance number of steps"
siabund,r,h,1.,0.,100.,"silicon abundance soft maximum"
siabundtyp,i,h,1,0,2,"silicon abundance variation type"
siabundint,i,a,1,0,1,"silicon abundance interpolation type"
siabundsof,r,a,0.,0.,1.,"silicon abundance soft minimum"
siabundnst,i,a,1,1,20,"silicon abundance number of steps"
sabund,r,h,1.,0.,100.,"sulfur abundance soft maximum"
sabundtyp,i,h,1,0,2,"sulfur abundance variation type"
sabundint,i,a,1,0,1,"sulfur abundance interpolation type"
sabundsof,r,a,0.,0.,1.,"sulfur abundance soft minimum"
sabundnst,i,a,1,1,20,"sulfur abundance number of steps"
arabund,r,h,1.,0.,100.,"argon abundance soft maximum"
arabundtyp,i,h,1,0,2,"argon abundance variation type"
arabundint,i,a,1,0,1,"argon abundance interpolation type"
arabundsof,r,a,0.,0.,1.,"argon abundance soft minimum"
arabundnst,i,a,1,1,20,"argon abundance number of steps"
caabund,r,h,1.,0.,100.,"calcium abundance soft maximum"
caabundtyp,i,h,1,0,2,"calcium abundance variation type"
caabundint,i,a,1,0,1,"calcium abundance interpolation type"
caabundsof,r,a,0.,0.,1.,"calcium abundance soft minimum"
```

```
caabundnst,i,a,1,1,20,"calcium abundance number of steps"
feabund,r,h,1.,0.,100.,"iron abundance soft maximum"
feabundtyp,i,h,1,0,2,"iron abundance variation type"
feabundint,i,a,1,0,1,"iron abundance interpolation type"
feabundsof,r,a,0.,0.,1.,"iron abundance soft minimum"
feabundnst,i,a,1,1,20,"iron abundance number of steps"
niabund,r,h,0.,0.,100.,"nickel abundance soft maximum"
niabundtyp,i,h,0,0,2,"nickel abundance variation type"
niabundint,i,a,1,0,1,"nickel abundance interpolation type"
niabundsof,r,a,0.,0.,1.,"nickel abundance soft minimum"
niabundnst,i,a,1,1,20,"nickel abundance number of steps"
spectrum,s,a,"pow",,,"spectrum type?"
spectrum_file,s,a,"spct.dat",,,"spectrum file?"
spectun,i,a,0,0,1,"spectrum units? (0=energy, 1=photons)"
redshift,i,h,1,0,1,"Is redshift a parameter? (0=no, 1=yes)"
nsteps,i,h,3,1,1000,"number of steps"
niter,i,h,0,,,"number of iterations"
lwrite,i,h,0,0,1,"write switch (1=yes, 0=no)"
lprint,i,h,0,0,1,"print switch (1=yes, 0=no)"
lstep,i,h,0,,,"step size choice switch"
npass,i,h,1,1,10000,"number of passes"
lcpres,i,h,0,0,1,"constant pressure switch (1=yes, 0=no)"
emult,r,h,1.,1.e-6,1.e+6,"Courant multiplier"
taumax,r,h,2.,1.,10000.,"tau max for courant step"
xeemin,r,h,1.e-6,1.e-6,0.5,"minimum electron fraction"
critf,r,h,1.e-14,1.e-24,0.1,"critical ion abundance"
vturbi,r,h,1.,0.,30000.,"turbulent velocity (km/s)"
modelname,s,a,"coronal grid",,,"model name"
```

```
loopcontrol,i,h,0,0,30000,"loop control (0=standalone)"
elow,r,h,1.0E+2,0.,5.11E+5,"energy band low end (eV)"
ehigh,r,h,2.0E+4,0.,5.11E+5,"energy band high end (eV)"
mode,s,h,"ql",,,"mode"
```

3.2. Example 2: Photoionized Grid

In this example a grid of photoionization models is calculated with varying ionization parameter, column density, and element abundances. The ionizing spectrum is a power law with index -1.

```
cfrac,r,a,0.,0.,1.,"covering fraction soft maximum"
cfractyp,i,h,0,0,2,"covering fraction variation type"
cfracint,i,a,1,0,1,"covering fraction interpolation type"
cfracsof,r,a,0.,0.,1.,"covering fraction soft minimum"
cfracnst,i,a,1,1,20,"covering fraction number of steps"
temperature,r,h,1.,0.,1.E4,"temperature soft maximum (/10**4K)"
temperaturetyp,i,h,0,0,2,"temperature variation type"
temperatureint,i,a,1,0,1,"temperature interpolation type"
temperaturesof,r,a,0.,0.,1.,"temperature soft minimum"
temperaturenst,i,a,1,1,20,"temperature number of steps"
pressure,r,h,0.03,0.,1.,"pressure soft maximum (dyne/cm**2)"
pressuretyp,i,h,0,0,2,"pressure variation type"
pressureint,i,a,1,0,1,"pressure interpolation type"
pressuresof,r,a,0.,0.,1.,"pressure soft minimum"
pressurenst,i,a,1,1,20,"pressure number of steps"
density,r,a,1.E+10,0.,1.E18,"density soft maximum (cm**-3)"
densitytyp,i,h,0,0,2,"density variation type"
```

```
densityint,i,a,1,0,1,"density interpolation type"
densitysof,r,a,0.,0.,1.,"density soft minimum"
densitynst,i,a,1,1,20,"density number of steps"
trad,r,a,-1.,,,"radiation temperature or alpha soft maximum?"
tradtyp,i,h,0,0,2,"radiation temperature variation type"
tradint,i,a,1,0,1,"radiation temperature interpolation type"
tradsof,r,a,0.,0.,1.,"radiation temperature soft minimum"
tradnst,i,a,1,1,20,"radiation temperature number of steps"
rlrad38,r,a,1.e+6,0.,1.E10,"luminosity soft maximum (/10**38 erg/s)"
rlrad38typ,i,h,0,0,2,"luminosity variation type"
rlrad38int,i,a,1,0,1,"luminosity interpolation type"
rlrad38sof,r,a,0.,0.,1.,"luminosity soft minimum"
rlrad38nst,i,a,1,1,20,"luminosity number of steps"
column,r,a,1.E23,0.,1.E25,"column density soft maximum (cm**-2)"
columntyp,i,h,2,0,2,"column density variation type"
columnint,i,a,1,0,1,"column density interpolation type"
columnsof,r,a,1.E20,1.,1.E25,"column density soft minimum"
columnnst,i,a,7,1,20,"column density number of steps"
rlogxi,r,a,4.5,-10.,+10.,"log(ionization parameter) soft maximum (erg cm/s)"
rlogxityp,i,h,2,0,2,"log(ionization parameter) variation type"
rlogxiint,i,a,0,0,1,"log(ionization parameter) interpolation type"
rlogxisof,r,a,-1.,-10.0,+10.0,"log(ionization parameter) soft minimum"
rlogxinst,i,a,20,1,20,"log(ionization parameter) number of steps"
habund,r,h,1.,0.,100.,"hydrogen abundance soft maximum"
habundtyp,i,h,0,0,2,"hydrogen abundance variation type"
habundint,i,a,1,0,1,"hydrogen abundance interpolation type"
habundsof,r,a,0.,0.,1.,"hydrogen abundance soft minimum"
habundnst,i,a,1,1,20,"hydrogen abundance number of steps"
```

```
heabund,r,h,1.,0.,100.,"helium abundance soft maximum"
heabundtyp,i,h,0,0,2,"helium abundance variation type"
heabundint,i,a,1,0,1,"helium abundance interpolation type"
heabundsof,r,a,0.,0.,1.,"helium abundance soft minimum"
heabundnst,i,a,1,1,20,"helium abundance number of steps"
cabund,r,h,1.,0.,100.,"carbon abundance soft maximum"
cabundtyp,i,h,1,0,2,"carbon abundance variation type"
cabundint,i,a,1,0,1,"carbon abundance interpolation type"
cabundsof,r,a,0.,0.,1.,"carbon abundance soft minimum"
cabundnst,i,a,1,1,20,"carbon abundance number of steps"
nabund,r,h,1.,0.,100.,"nitrogen abundance soft maximum"
nabundtyp,i,h,1,0,2,"nitrogen abundance variation type"
nabundint,i,a,1,0,1,"nitrogen abundance interpolation type"
nabundsof,r,a,0.,0.,1.,"nitrogen abundance soft minimum"
nabundnst,i,a,1,1,20,"nitrogen abundance number of steps"
oabund,r,h,1.,0.,100.,"oxygen abundance soft maximum"
oabundtyp,i,h,1,0,2,"oxygen abundance variation type"
oabundint,i,a,1,0,1,"oxygen abundance interpolation type"
oabundsof,r,a,0.,0.,1.,"oxygen abundance soft minimum"
oabundnst,i,a,1,1,20,"oxygen abundance number of steps"
neabund,r,h,1.,0.,100.,"neon abundance soft maximum"
neabundtyp,i,h,1,0,2,"neon abundance variation type"
neabundint,i,a,1,0,1,"neon abundance interpolation type"
neabundsof,r,a,0.,0.,1.,"neon abundance soft minimum"
neabundnst,i,a,1,1,20,"neon abundance number of steps"
mgabund,r,h,1.,0.,100.,"magnesium abundance soft maximum"
mgabundtyp,i,h,1,0,2,"magnesium abundance variation type"
mgabundint,i,a,1,0,1,"magnesium abundance interpolation type"
```

```
mgabundsof,r,a,0.,0.,1.,"magnesium abundance soft minimum"
mgabundnst,i,a,1,1,20,"magnesium abundance number of steps"
siabund,r,h,1.,0.,100.,"silicon abundance soft maximum"
siabundtyp,i,h,1,0,2,"silicon abundance variation type"
siabundint,i,a,1,0,1,"silicon abundance interpolation type"
siabundsof,r,a,0.,0.,1.,"silicon abundance soft minimum"
siabundnst,i,a,1,1,20,"silicon abundance number of steps"
sabund,r,h,1.,0.,100.,"sulfur abundance soft maximum"
sabundtyp,i,h,1,0,2,"sulfur abundance variation type"
sabundint,i,a,1,0,1,"sulfur abundance interpolation type"
sabundsof,r,a,0.,0.,1.,"sulfur abundance soft minimum"
sabundnst,i,a,1,1,20,"sulfur abundance number of steps"
arabund,r,h,1.,0.,100.,"argon abundance soft maximum"
arabundtyp,i,h,1,0,2,"argon abundance variation type"
arabundint,i,a,1,0,1,"argon abundance interpolation type"
arabundsof,r,a,0.,0.,1.,"argon abundance soft minimum"
arabundnst,i,a,1,1,20,"argon abundance number of steps"
caabund,r,h,1.,0.,100.,"calcium abundance soft maximum"
caabundtyp,i,h,1,0,2,"calcium abundance variation type"
caabundint,i,a,1,0,1,"calcium abundance interpolation type"
caabundsof,r,a,0.,0.,1.,"calcium abundance soft minimum"
caabundnst,i,a,1,1,20,"calcium abundance number of steps"
feabund,r,h,1.,0.,100.,"iron abundance soft maximum"
feabundtyp,i,h,1,0,2,"iron abundance variation type"
feabundint,i,a,1,0,1,"iron abundance interpolation type"
feabundsof,r,a,0.,0.,1.,"iron abundance soft minimum"
feabundnst,i,a,1,1,20,"iron abundance number of steps"
niabund,r,h,0.,0.,100.,"nickel abundance soft maximum"
```

```
niabundtyp,i,h,0,0,2,"nickel abundance variation type"
niabundint,i,a,1,0,1,"nickel abundance interpolation type"
niabundsof,r,a,0.,0.,1.,"nickel abundance soft minimum"
niabundnst,i,a,1,1,20,"nickel abundance number of steps"
spectrum,s,a,"pow",,,"spectrum type?"
spectrum_file,s,a,"spct.dat",,,"spectrum file?"
spectun,i,a,0,0,1,"spectrum units? (0=energy, 1=photons)"
redshift,i,h,1,0,1,"Is redshift a parameter? (0=no, 1=yes)"
nsteps,i,h,3,1,1000,"number of steps"
niter,i,h,99,,,"number of iterations"
lwrite,i,h,0,0,1,"write switch (1=yes, 0=no)"
lprint,i,h,0,0,1,"print switch (1=yes, 0=no)"
lstep,i,h,0,,,"step size choice switch"
npass,i,h,1,1,10000,"number of passes"
lcpres,i,h,0,0,1,"constant pressure switch (1=yes, 0=no)"
emult,r,h,0.5,1.e-6,1.e+6,"Courant multiplier"
taumax,r,h,5.,1.,10000.,"tau max for courant step"
xeemin,r,h,0.1,1.e-6,0.5,"minimum electron fraction"
critf,r,h,1.e-14,1.e-24,0.1,"critical ion abundance"
vturbi,r,h,1.,0.,30000.,"turbulent velocity (km/s)"
modelname,s,a,"photoionized grid",,,"model name"
loopcontrol,i,h,0,0,30000,"loop control (0=standalone)"
elow,r,h,1.0E+2,0.,5.11E+5,"energy band low end (eV)"
ehigh,r,h,2.0E+4,0.,5.11E+5,"energy band high end (eV)"
mode,s,h,"ql",,,"mode"
```

Chapter 8

WARMABS

An alternative method for fitting XSTAR results to observed spectra within xspec is to use the xspec 'analytic' model warmabs. This actually includes several separate models: warmabs, photemis, hotabs, hotemis, windabs and multabs. These are described below.

This model allows the use of xspec models for warm absorbers and photoionized emitters, and for coronal equilibrium absorbers and emitters without requiring construction of mtables or etables. The advantages of this procedure include:

1) Circumventing the intrinsic approximations associated with use of tables for absorption with variable abundances treated as multiplicative parameters. (see the 'Important Notes on Mtables' section of the xstar2xspec chapter of the xstar manual). Warmabs/photemis/windabs/multabs calculate spectra using stored level populations which are then scaled using element abundances specified during the xspec session before the spectra are calculated. Therefore the approximations associated with the use of tables are avoided. (However, see the section below on the current limitations of these models).

2) Circumventing the intrinsic clumsiness of the use of tables.

3) Ability to use arbitrary spectral resolution, not limited by the internal xstar spectral resolution

3.5) Allows the use of turbulent broadening as a fitting parameter in xspec.

4) This model employs the most recent updates to xstar and the database, version 2.2.

1. Obtaining warmabs

As of this writing, the warmabs package is not included as part of the standard xstar distribution, either within heasoft or as part of standalone xstar. Instead, it must be downloaded and installed separately from the ftp site: ftp://legacy.gsfc.nasa.gov/software/plasma_codes/xstar/warmabs22.tar.gz

The contents of the tarfile include:

atdb.fits – xstar atomic database binary fits file. This must reside in directory pointed to by the WARMABS_DATA environment variable. This file is the same as that used by xstar version 2.2.

coheat.dat – compton heating/cooling data file. This must reside in directory pointed to by WARMABS_DATA environment variable. This file is identical to the file coheat.dat which is part of xstar.

fphotems.f – source code for warmabs, photemis, windabs, multabs, hotemis, and hotabs

lmodel_warmabs.dat – local model definition file needed by xspec. When using xspec11, must be renamed to lmodel.dat or catted onto existing lmodel.dat

pops.fits.nxx binary fits files containing pre-calculated level populations for use with the warmabs or photemis model. Currently the user chooses which of these to use using the WARMABS_POP environment variable, eg., by doing 'setenv

WARMABS_POP pops.fits.n4' (note that the full path is not used here, rather the path relative to the WARMABS_DATA path).

popshot.fi – binary fits file containing pre-calculated level populations for the hotabs and hotemis models. This file also must reside in directory pointed to by WARMABS_DATA environment variable. This file is not intended for modification by the user

instructions.txt – text version of the xspec11 manual page describing the setup of analytic models

xautosav.xcm – sample xspec script file showing use of model.

README – this file

2. Installation

The procedure for setting up and using this model is as described in the xspec manual:

0) You need to have the heasoft package installed on your machine, but it must be built from source. Local models cannot be installed from a binary installation.

1) untar this directory somewhere in your user area

2) setup your headas environment (eg. 'setenv HEADAS /path/to/architecture', and 'source $HEADAS/headas-init.csh')

3) point the WARMABS_DATA enviroment variable to the directory where the xstar data files atdb.fits, coheat.dat and pops.fits are kept (eg, 'setenv WARMABS_DATA $PWD'). This must be done AFTER the environment variables for xspec are set up.

4) In addition the WARMABS_POP environment variable must be set to point

to the name of the populations file (relative to the WARMABS_DATA location) to be used by the warmab, windabs, multabs and photemis models. For example, the populations for a density 10^4 gas illuminated by a Γ=2 power law are in pops.fits.n04, so to use these do 'setenv WARMABS_POP pops.fits.n04'.

5) Start up xspec, and in response to the prompt type 'initpackage xstarmod lmodel_warmabs.dat ¡path-to-current-directory¿', where ¡path-to-current-directory¿ is the full path of the current directory. Then, after the build is complete type 'lmod xstarmod ¡path-to-current-directory¿' In subsequent sessions you don't neet to do the initpackage step again, just the lmod.

6) Instructions for xspec11:

a) Set the environment variable LMODDIR to the local directory, where you have untarred the package, eg. 'setenv LMODDIR $PWD'

b) make sure this directory is pointed to by your shared object library environment variable, as described in the xspec11 manual, eg. 'setenv LD_LIBRARY_PATH "*LMODDIR* :LD_LIBRARY_PATH"'

c) rename the lmodel_warmabs.dat file to lmodel.dat, or cat it onto an existing lmodel.dat

d) cd into $LHEASOFT/../spectral/xspec/src/local_mod

e) type hmake

f) cd to the directory where you want to work and start xspec 11.

3. warmabs

Inside of xspec, the model can be invoked by typing 'mo warmabs*pow' or variations on that. The input parameters include: absorber column, parameterized as log(N/10^{22} cm^{-2}), log(ionization parameter) element abundances relative to solar, turbulent broadening in km/s, and redshift.

4. photemis

The photoionized emitter can be invoked by typing 'mo photemis'. This model is the 'thermal' (i.e. recombination and collisional excitation) emission which comes from the same plasma used in warmabs. Note that this does NOT include the resonant scattered emission associated with the warmabs line emission; the windabs model can be used, with a trick, to model this emission if desired (see below).

The model supplies to xspec the emissivity of the gas, in units of erg cm^{-3} s^{-1}, times a factor 10^{10}. So the physical meaning of the normalization, κ, is

$$\kappa = \frac{\mathrm{EM}}{4\pi D^2} \times 10^{-10} \tag{8.1}$$

where EM is the emission measure of the gas in the source (at the ionization parameter used in the fit) and D is the distance to the source. This same relation applies to the results from the hotabs model.

5. windabs

Windabs uses the 'sei' method of Lamers et al. 1987 Ap. J. 314 726 to take the line center optical depth calculated by the usual warmabs routines and spread them in wavelength space in order to model the scattering in an outflowing wind. Bound-free absorption is treated the same as in warmabs. For lines, the optical depths from warmabs are input to the sei routines as the value of the paramter ttot (optical depth normalization). The input velocity vturb is used as the terminal velocity of the wind. Other sei parameters are assumed to be fixed: gamma=1 (run of optical depth with velocity), apha=1 (velocity law), and eps=0 (thermalization parameter). Windabs has the same input parameters as warmabs, except that it also has a covering fraction parameter, C (not part of the standard sei formulation), which should be in the range 0-2. The physical meaning of C is that, when $0 \leq C \leq 1$, the emission component is reduced by a factor of C. A feature added on 09/04/2007: when covering fraction $C \geq 1$, the absorption component is reduced by a factor $2 - C$. So at $C = 2$, the model produces pure scattered emission. Values of $C \geq 2$ have no physical meaning.

6. multabs

Multabs tries to account for line broadening by absorption by multiple discrete components rather than by turbulence or bulk flow. This model is essentially identical to warmabs in that it uses the warm absorber spectrum generated by warmabs, but initially assuming that the lines are broadened only by thermal gas motions. It then replicates these lines a fixed number of lines and spreads the components over a given velocity width. The input parameters include the same parameters as warmabs: ionization parameter, column, and abundances. These control the properties of the individual absorbing components, in the same way as for warmabs. In addition, the user specifies the velocity spread of the components, still called vturb, and the covering fraction, cfrac. This covering fraction can be interpreted as a covering fraction in velocity space. The number of discrete components is given by cfrac*vturb/vtherm, where vturb is input by the user and vtherm is the thermal line width which is determined by the equilibrium temperature (calculated by xstar). The optical depth of each component is divided by the number of components, so that the total optical depth summed over the components is independent of their number. The number of components cannot be less than 1. If it is 1 then the component will be placed at the redshifted energy of the line. If it is greater than 1, then the components will be spaced uniformly in velocity from -vturb to +vturb relative to the rest energy of the line. There is no restriction on the value of cfrac, so it is possible to set cfrac to some large number and thereby fill a uniform trough in velocity with the line.

7. hotabs

Hotabs and hotemis are the coronal analogs of warmabs and photemis. In this case the free parameter determining the ionization is the log of the temperature in units of 10^4 K, i.e. logt4=0 corresponds to 10^4 K, logt4=3 corresponds to 10^7K.

The normalization is the same as for warmabs. The model supplies to xspec the emissivity of the gas, in units of erg cm^{-3} s^{-1}, times a factor 10^{10}. So the physical meaning of the normalization, κ, is

$$\kappa = \frac{\mathrm{EM}}{4\pi D^2} \times 10^{-10} \tag{8.2}$$

where EM is the emission measure of the gas in the source (at the temperature used in the fit) and D is the distance to the source.

8. Creating Your Own pops.fits Files

The procedure for doing this is as follows:

a) You must have xstar version 2.2 installed, either as standalone or as part of the heasoft distribution. Earlier versions cannot be used to generate population files for use by warmabs and photemis.

b) Using xstar version 2.2, run a variation of the constant density sphere described in the manual chapter 2, with low density and low luminosity so that the sphere is optically thin. warmabs and photemis treat the ionization parameter as the free parameter describing level populations, so it is desirable that the model run in xstar be optically thin, and that it span the range of ionization parameter of interest. An example of the xstar command run from the command line is:

xstar cfrac=0. temperature=10000. pressure=0.03 density=10000. spectrum='pow' trad=-1. rlrad38=1.e-10 column=1.e+17 rlogxi=5. lcpres=0 habund=1. heabund=1. cabund=1. nabund=1. oabund=1. neabund=1. mgabund=1. siabund=1. sabund=1. arabund=1. caabund=1. feabund=1. niabund=0. modelname="otfs" niter=99 npass=1 critf=1.e-7 nsteps=10 xeemin=0.04 emult=0.1 taumax=50. lprint=1 lwrite=1

and an example of the xstar.par file which could be used instead is:

```
cfrac,r,a,1,0.,1.,"covering fraction"
temperature,r,a,10000,0.,1.e4,"temperature (/10**4K)"
lcpres,i,a,0,0,1,"constant pressure switch (1=yes, 0=no)"
pressure,r,a,0.03,0.,1.,"pressure (dyne/cm**2)"
density,r,a,1.e+4,0.,1.e18,"density (cm**-3)"
spectrum,s,a,"pow",,,"spectrum type?"
spectrum_file,s,a,"spct.dat",,,"spectrum file?"
spectun,i,a,0,0,1,"spectrum units? (0=energy, 1=photons)"
trad,r,a,-1,,,"radiation temperature or alpha?"
rlrad38,r,a,1.00E-015,0.,1.e10,"luminosity (/10**38 erg/s)"
column,r,a,1.00E+016,0.,1.e25,"column density (cm**-2)"
rlogxi,r,a,5,-10.,+10.,"log(ionization parameter) (erg cm/s)"
habund,r,a,1,0.,100.,"hydrogen abundance"
heabund,r,a,1,0.,100.,"helium abundance"
liabund,r,h,0,0.,100.,"lithium abundance"
beabund,r,h,0,0.,100.,"beryllium abundance"
babund,r,h,0,0.,100.,"boron abundance"
cabund,r,a,1,0.,100.,"carbon abundance"
nabund,r,a,1,0.,100.,"nitrogen abundance"
oabund,r,a,1,0.,100.,"oxygen abundance"
```

```
fabund,r,a,1,0.,100.,"fluorine abundance"
neabund,r,a,1,0.,100.,"neon abundance"
naabund,r,a,1,0.,100.,"sodium abundance"
mgabund,r,a,1,0.,100.,"magnesium abundance"
alabund,r,a,1,0.,100.,"aluminum abundance"
siabund,r,a,1,0.,100.,"silicon abundance"
pabund,r,a,1,0.,100.,"phosphorus abundance"
sabund,r,a,1,0.,100.,"sulfur abundance"
clabund,r,a,1,0.,100.,"chlorine abundance"
arabund,r,a,1,0.,100.,"argon abundance"
kabund,r,a,1,0.,100.,"potassium abundance"
caabund,r,a,1,0.,100.,"calcium abundance"
scabund,r,a,1,0.,100.,"scandium abundance"
tiabund,r,a,1,0.,100.,"titanium abundance"
vabund,r,a,1,0.,100.,"vanadium abundance"
crabund,r,a,1,0.,100.,"chromium abundance"
mnabund,r,a,1,0.,100.,"manganese abundance"
feabund,r,a,1,0.,100.,"iron abundance"
coabund,r,a,1,0.,100.,"cobalt abundance"
niabund,r,a,1,0.,100.,"nickel abundance"
cuabund,r,a,1,0.,100.,"copper abundance"
znabund,r,a,1,0.,100.,"zinc abundance"
modelname,s,a,"otfs",,,"model name"
nsteps,i,h,3,1,1000,"number of steps"
niter,i,h,0,,,"number of iterations"
lwrite,i,h,0,0,1,"write switch (1=yes, 0=no)"
lprint,i,h,0,0,2,"print switch (1=yes, 0=no)"
lstep,i,h,0,,,"step size choice switch"
```

```
emult,r,h,0.5,1.e-6,1.e+6,"Courant multiplier"

taumax,r,h,5.,1.,10000.,"tau max for courant step"

xeemin,r,h,0.1,1.e-6,0.5,"minimum electron fraction"

critf,r,h,1.e-7,1.e-24,0.1,"critical ion abundance"

vturbi,r,h,1.,0.,30000.,"turbulent velocity (km/s)"

radexp,r,h,0.,-3.,3.,"density distribution power law index"

ncn2,i,h,999,999,99999,"number of continuum bins"

loopcontrol,i,h,0,0,30000,"loop control (0=standalone)"

npass,i,h,1,1,10000,"number of passes"

mode,s,h,"ql",,,"mode"
```

It is important to have the write switch set to 1 in order to generate the file containing the level populations at each step.

In doing this, you may want to change the shape of the ionizing spectrum. In these examples it is a power law with γ=2, which is the same as what is used in the distributed warmabs/photemis package. The maximum ionization parameter is controlled by the value of the variable rlogxi, in this case it is 5, corresponding to ξ=1.e5. The minimum ionization parameter is set by the column density of the model; this input file will terminate when log(ξ) falls below -2. The number of intermediate steps is controlled by the variable nsteps; in this case nsteps=6 corresponds to a uniform spacing of 0.13 in log(ξ). If the final model has more than 200 spatial zones, warmabs and photemis will discard any populations for spatial zones beyond 200.

c) After the run is complete, the populations are in the file xout_detail.fits. This must be moved to the directory containing the xstar data files, i.e. the directory pointed to by the WARMABS_DATA environment variable, and then pointed to by the WARMABS_POP environment variable. Then warmabs and photemis will read populations from this file. The file lmodel.dat, as it is distributed, contains limits on the ionization parameter which may not be appropriate to your file pops.fits, so you may want to change this.

9. Common block 'ewout'

A feature added September 2007 is output of the strongest lines, sorted by element and ion into a common block called 'ewout' This feature is only available for the warmabs, photemis, hotemis, hotabs models (not windabs, or multabs). The contents of the common block are:

lmodtyp: identifies which model most recently put its output into the common block. lmodtyp=1,2,3,4, where 1=hotabs, 2=hotemis, 3=warmabs, 4=photemis

newout: number of lines in the list. This is zeroed after each call to warmabs, photemis, hotabs, etc.

lnewo: array conatining line indexes. These should correspod to the line indexes in the ascii line lists on the xstar web page.

kdewo: character array containing the name of the ion

kdewol: character array containing the name of the lower level

kdewou: character array containing the name of the upper level

aijewo: array containing A values for the lines

flinewo: array containing f values for the lines

ggloewo: array containing statistical weights for the lower levels

ggupewo: array containing statistical weights for the upper levels

elewo: array containing the line wavelengths

tau0ewo: array containing the line center depths

tau02ewo:array containing the line depths at the energy bin nearest to line center ewout: array containing line equivalent widths in eV, negative values correspond to emission

elout: array containing line luminosities in xstar units (erg/s/10$\hat{3}$8)

The details of how to get at the contents of the common block are up to the user. Currently xspec does not have a mechanism to do this, but it is straightforward to write a small fortran code to call the models with suitable parameter values and print the common block from there. The calling sequence for an analytic model is described in the xspec manual. It is important to point out that the common block is overwritten at each call to one of the models, so it should be emptied by the calling program after each call to one of the models.

An example is as follows:

```
      program fphottst
c
      implicit none
c
      real ear(0:20000),photar(20000),photer(10000),param(30)
      integer ne,mm,ifl
      real emin,emax,dele
c
      ne=10000
      emin=0.4
      emax=7.2
      dele=(emax/emin)**(1./float(ne-1))
      ear(0)=emin
      do mm=1,ne
        ear(mm)=ear(mm-1)*dele
        enddo
```

```
      write (6,*)ear(1),ear(ne),ear(ne/2)
      param(1)=2.
      param(2)=-4.
      param(13)=100.
      param(12)=0.
      param(3)=1.
      param(4)=1.
      param(5)=1.
      param(6)=1.
      param(7)=1.
      param(8)=1.
      param(9)=1.
      param(10)=1.
      param(11)=1.
      param(3)=0.
      call fhotabs(EAR,NE,PARAM,IFL,PHOTAR,PHOTER)
c
      call commonprint
c
      write (6,*)'after fwarmabs'
      do mm=1,ne
        write (6,*)ear(mm),photar(mm)/ear(mm)
        enddo
c
      stop
      end
      subroutine commonprint
c
```

```
      implicit none  !jg
c
      parameter (nnnl=200000)
c
      common /ewout/newout,lnewo(nnnl),kdewo(8,nnnl),
     $  kdewol(20,nnnl),kdewou(20,nnnl),aijewo(nnnl),flinewo(nnnl),
     $  ggloewo(nnnl),ggupewo(nnnl),
     $  elewo(nnnl),tau0ewo(nnnl),tau02ewo(nnnl),ewout(nnnl),
     $  elout(nnnl),lmodtyp
c
      real aijewo,flinewo,ggloewo,ggupewo,elewo,tau0ewo,tau02ewo,
     $  ewout,elout
      integer lnewo,newout,lmodtyp
      character*1 kdewo,kdewol,kdewou
      integer kk,mm   !jg
c
       if (lmodtyp.eq.1) write (6,*)'after hotabs',newout
       if (lmodtyp.eq.2) write (6,*)'after hotemis',newout
       if (lmodtyp.eq.3) write (6,*)'after warmabs',newout
       if (lmodtyp.eq.4) write (6,*)'after photemis',newout
       write (16,*)'index, ion, wave(A), tau0, tau0grid, ew (eV),',
     $ 'lum, lev\_low, lev\_up, a\_ij, f\_ij, g\_lo, g\_up'
        do kk=1,newout
         write (16,9955)kk,lnewo(kk),(kdewo(mm,kk),mm=1,8),
     $     elewo(kk),tau0ewo(kk),tau02ewo(kk),ewout(kk),
     $     elout(kk),
     $     (kdewol(mm,kk),mm=1,20),(kdewou(mm,kk),mm=1,20),
     $     aijewo(kk),flinewo(kk),ggloewo(kk),ggupewo(kk)
```

```
         enddo
9955   format (1x,2i8,1x,8a1,5(1pe11.3),1x,2(20a1,1x),4(1pe11.3))
c
      return
      end
```

10. Limitations

This package is still being tested. Some embarrassing bugs have already been found, but more may still lurk. Please contact me with any reports or questions.

2) It is not blindingly fast. On a 1GHz machine, the first time 'mo warmabs ..' is typed, the initial setup requires ∼30 seconds. After that, each time an abundance or ionization parameter is changed, it requires 5-10 seconds to recalculate the model. Presumably on a faster machine this will be reduced.

3) It calculates the spectrum 'on the fly', appropriate to the energy grid and parameters the user specifies. But it does not calculate the ionization balance self-consistently. It uses a saved file of level populations calculated for a grid of optically thin models calculated with a $\Gamma = 2$ power law ionizing spectrum. So this will not be self-consistent if your source has a very different ionizing spectrum. Also it implicitly assumes that the absorber has uniform ionization even if you specify a large column, which is not self-consistent.

4) Photemis does not take into account scattering, only true emission.

Chapter 9

The Physics Behind XSTAR

A. Assumptions

In this section we describe the computational procedure, assumptions, free parameters, and the quantities which are calculated. Chief among the assumptions is that each model consists of a spherical gas cloud with a point source of continuum radiation at the center. Therefore it implicitly assumes spherical symmetry and radially beamed incident radiation. In principle, more complicated geometries can be mimiced by adding the local emission from various spherical sections with appropriately chosen conditions. Also important is the assumption that all physical processes affecting the state of the gas are in a steady-state, i.e. that the the timescales for variation in the gas density and illuminating radiation are long compared with timescales affecting all atomic processes and propogation of radiation within the gas. The validity of this assumption in any given situation depends on the conditions there, such as the gas density, temperature, and degree of ionization, and can be evaluated by using a model assuming steady-state and then calculating atomic rates which can (hopefully) justify the steady-state assumption a posteriori.

The primary difference between these models and atmospheric models lies in the treatment of the radiation field. In an optically thick atmosphere the state of the gas at any point in the cloud is coupled to the state of the gas in a large part of the rest of the cloud by the continuum radiation field and, in the limit of very large optical depth, can affect the excitation and ionization by suppressing radiative free-bound (recombination) transitions. We attempt to mimic some of these effects by assigning to each recombination event an escape probability, using an expression given in the following section. We also calculate the transfer of radiation by assuming that diffuse radiation emitted at each radius is directed radially outward or inward. These assumptions will be described in more detail later in this section.

A further assumption governs the treatment of the transport of radiation in spectral lines. Over a wide range of plausible situations large optical depths occur in the cores of lines of abundant ions, which may be important in cooling the gas. In treating the transfer of these photons we make the (conventional) assumption of complete redistribution in the scattering, which assumes that the transfer of the line photons occurs in a spatial region very close to the point where the photons are emitted. Therefore the line emission rates are multiplied by an escape probability using an expression given in the following section. This factor is intended to simulate the line scattering in the immediate vicinity of the emission region, and it assumes that escape from this region occurs when the photon scatters into a frequency where the optical depth is less than unity. Following escape from the local region, the line photon is assumed to be subject to absorption by continuum processes which are treated using the same 2-stream transfer equation as for the continuum.

B. Input

The input parameters are the source spectrum, the gas composition, and the gas density or pressure. The spectrum of the central source of radiation is described by the spectral luminosity, $L_{0\varepsilon} = Lf_\varepsilon$, where L is the total luminosity (in erg s^{-1}). The spectral function, f_ε, is normalized such that $\int_0^\infty f_\varepsilon d\varepsilon = 1$ and may be of one of a variety of types, including: Thermal bremsstrahlung, $f_\varepsilon \sim \exp(-\varepsilon/kT)$; blackbody, $f_\varepsilon \sim \varepsilon^3/[\exp(\varepsilon/kT) - 1]$; or power law, $f_\varepsilon \sim \varepsilon^\alpha$; or the user may define the form of the ionizing continuum by providing a table of energies and fluxes. The gas consists of the elements H, He, C, N, O, Ne, Mg, Si, S, Ar, Ca, Fe, and Ni with relative abundances specified by the user. The default abundances are the solar values given by Grevesse, Noels and Sauval 1996.

C. Elementary Considerations

When the gas is optically thin, the radiation field at each radius is determined simply by geometrical dilution of the given source spectrum f_ε. Then, as shown by Tarter Tucker and Salpeter 1969, the state of the gas depends only on the ionization parameter $\xi = L/nR^2$, where L is the (energy) luminosity of the incident radiation integrated from 1 to 1000 Ry, n is the gas density, and R is the distance from the radiation source. This scaling law allows the results of one model calculation to be applied to a wide variety of situations. For a given choice of spectral shape this parameter is proportional to the various other customary ionization parameter definitions, i.e. $U_H = F_H/n$ (Davidson and Netzer 1979), where F_H is the incident photon number flux above 1 Ry; $\Gamma = F_\nu(\nu_L)/(2hcn)$, where $F_\nu(\nu_L)$ is incident (energy) flux at 1 Ry; and $\Xi = L/(4\pi R^2 cnkT)$ (e.g. Krolik McKee and Tarter, 1981).

In the optically thick case, Hatchett Buff and McCray 1976, and Kallman 1983 showed that the state of the gas could be parameterized in terms of an additional parameter which is a function of the product of L and either n (the number density) or P (the pressure), depending on which quantity is held fixed. In the case $n =$ constant, this second parameter is simply $(Ln)^{1/2}$ (McCray, Wright and Hatchett 1977). This parameter does not allow easy scaling of model results from value of Ln to another, since the dependence on this parameter is non-linear, but it does provide a useful indicator of which combinations of parameter values are likely to yield similar results and vice versa.

When the electron scattering optical depth, τ_e, of the cloud becomes significant, the outward-only approximation used here breaks down, and different methods of describing the radiative transfer must be used (e.g. Ross 1979). Therefore, the range of validity of the models presented here is restricted to $\tau_e \leq 0.3$, or electron column densities $\leq 10^{24}$ cm^{-2}.

D. Algorithm

The construction of a model consists of the simultaneous determination of the state of the gas and the radiation field as a function of distance from the source.

D.1. Atomic Level Populations

The state of the gas is defined by its temperature and by the ionic level populations. As a practical matter, we maintain the distinction between the total abundance of a given ion relative to its parent element (the ion fraction or fractional abundance) and the relative populations of the various bound levels of that ion (level populations), although such distinctions are somewhat arbitrary given the presence of transitions linking non-adjacent ion stages and excited levels of adjacent ions.

Calculation of level populations proceeds in 2 steps. First, a calculation of ion fractions is performed using total ionization and recombination rates into and out of each ion analogous to those used in XSTAR v.1 (KM82). Then we eliminate ions with abundance less than a fixed fraction ϵ relative to hydrogen from further consideration. Experimentation has shown that $\epsilon = 10^{-8}$ (as parameterized by the input parameter critf) yields gas temperatures within 1% of those calculated using a larger set of ions for most situations when the density is low. This criterion leads to ion sets which can include up to 10 stages for heavy elements such as iron and nickel. We also make sure that the selected ions are all adjacent. i.e. we force the inclusion of ions which fall below our threshold if they are bracketed by ions which satisfy the abundance crterion.

The second step consists of solving the full kinetic equation matrix linking the various levels of the ions selected in step 1. We include all processes in the database which link the bound levels of any ion in our selected set with any other level, and also including the bare nucleus as the continuum level for the hydrogenic ion, if indicated. This results in a matrix with dimensionality which may be as large as 2400. The equations may be written schematically as $(rate\ in) = (rate\ out)$ for each level. In place of the equation for the ground level of the most abundant ion we solve the number conservation constraint.

We include collisional and radiative bound-bound transitions (with continuum photoexcitation), collisional ionization, photoionization and recombination for all the levels of every ion for which the required atomic rate data is available. The effects of line scattering in all transitions are accounted for by taking into account the fact that line scattering reduces the net decay rate by repeated absorption and reemission of the line photon. An analogous procedure is used for free-bound (recombination) transitions.

D.2. Atomic Levels

A large fraction of recombinations occur following cascades from a very large number of levels close to the continuum. Since explicit treatment of these levels is not feasible, we treat this process as follows (this procedure, along with detailed descriptions of other aspects of the database and the multilevel scheme are described in detail in Bautista and Kallman 2000): For every ion we choose a set of spectroscopic levels starting with the ground level, which are responsible for the identifiable emission lines and recombination continua; there are typically 10 – 50 of these for most ions, although for a few ions we include ≥100 such levels. In addition we include one or more superlevels and continuum levels. The continuum levels represent bound levels of more highly ionized species (in practice at most only a few such levels are of importance). The superlevel is an artificial level used to account for recombination onto the infinite series of levels that lie above the spectroscopic levels. In H and He-like ions the superlevels also account for the recombination cascades of these high lying levels onto the spectroscopic levels, and the rates for such decays are calculated by fitting to the results of population kinetic calculations for individual ions which explicitly include ≥1000 levels. For these isoelectronic sequences we explicitly include excited levels with a spectator electron, which give rise to satellite lines, excitation of these levels accounts for excitation-autoionization and radiative deexcitation, and recombination accounts for the dielectronic recombination process. For other iso-electronic sequences, the superlevels are assumed to decay directly to the ion's ground level, and the rates into and out of the superlevel are calculated in order to fit to the total recombination rates for the various ions Bautista and Kallman 2000. This approach allows us to simultaneously account for the contributions of excitation, ionization, and recombination to the ion's level populations. In this way we solve ionization and excitation balance without the use of total recombination rates which is customary in many nebular calculations.

By using the approach described above and providing that every transition process accompanied by its detailed balance inverse process we insure that the level populations will naturally converge to LTE under proper conditions.

D.3. Thermal Equilibrium

The temperature is found by solving the equation of thermal equilibrium, which may be written schematically as $(Heating) = (Cooling)$. This is solved simultaneously with the condition of charge conservation. We treat heating and cooling by calculating the rate of removal or addition of energy to local radiation field associated with each of the processes affecting level populations (this is in contrast to the method where these were calculated via their effects on the electron thermal bath as in KM82). Heating therefore includes photoionization heating and Compton heating. The cooling term includes radiative recombination, bremsstrahlung, and radiative deexcitation of bound levels. Cooling due to recombination and radiative deexcitation is included only for the escaping fraction, as described elsewhere in this section.

In the most highly ionized regions of our models, the dominant heating process is electron recoil following Compton scattering. In the non-relativistic approximation the net heating rate may be written (Ross 1979)

$$n_e\Gamma_e = \frac{\sigma_T}{m_e c^2}\left(\int \varepsilon J_\varepsilon d\varepsilon - 4kT\int J_\varepsilon d\varepsilon\right) \qquad (1)$$

Here σ_T is the Thomson cross section, n_e is the electron number density, T is the electron temperature, and J_ε is the local mean intensity in the radiation field. The first term in the brackets represents the heating of electrons by the X-rays, and the second term represents cooling of hot electrons by scattering with low energy photons. New in version 2.1e is a full relativistic treatment of Compton heating and cooling, and the addition of photon energy bins extending to 1 MeV. The energy shift per scattering is calculated by interpolating in a table (coheat.dat) which was calculated using the reults given by Guilbert (1986).

The spectrum of photoelectron energies for each ion is found by convolving the radiation field, weighted by photoelectron energy, with the photoionization cross section (see, e.g., Osterbrock 1974). The integral of this quantity provides the photoelectric heating rate.

The cooling rate due to radiative recombination is calculated by explicitly evaluating the quadrature over the recombination continuum spectrum for each recombining level, weighted by the escape fraction for that transition. The bremsstrahlung cooling rate is (Osterbrock 1974)

$$n_e\Gamma_e = 1.42 \times 10^{-27}T^{1/2}z^2n_en_z \text{ ergs cm}^{-3}\text{s}^{-1}, \tag{2}$$

where T is the electron temperature, is the electron number density, z is the charge on the cooling ion, and n_z is the ion density.

E. Recombination Continuum Emission and Escape

In analogy with the line emission, recombination emission and cooling rates are calculated using the continuum level population n_∞ and the quantities calculated from the photoionization cross section and the Milne relation. The spontaneous recombination rates are given by

$$\alpha_i = (\frac{n_i}{n_{i+1}n_e})^* \int_{\varepsilon_{th}}^{\infty} \frac{d\varepsilon}{\varepsilon}\frac{\varepsilon^3}{h^3c^2}\sigma_{pi}e^{(\varepsilon_{th}-\varepsilon)/kT} \tag{3}$$

where ${n_i}^*$ is the LTE density of ion i, and n_e is the electron density. The continuum emissivity due to this process is given by

$$j_\varepsilon = n_{upper}n_e(\frac{n_i}{n_{i+1}n_e})^*\frac{\varepsilon^3}{h^3c^2}\sigma_{pi}e^{(\varepsilon_{th}-\varepsilon)/kT} \tag{4}$$

where n_{upper} is the number density of ions in the recombining level and σ_{pi} is the photoionization cross section. The cooling rate is given by the integral of this expression over energy. These rates are calculated separately for each level included in the multilevel calculation.

In order to account for the suppression of rates due to emission and reabsorption of recombination continua, we multiply the rates and emissivities by an escape fraction given by:

$$P_{esc.,cont.} = \frac{1}{1000\tau_{cont.} + 1} \tag{5}$$

where $\tau_{cont.}$ is the optical depth at the threshold energy for the relevant transition. This factor is used to correct both the emission rate for the recombination events, and also the rates in the kinetic equations determining level populations etc, and has been found to give reasonably good fits to the results of more detailed calculations for the case of H II region models in which the Lyman continuum of hydrogen is optically thick (e.g. Harrington 1989).

E.1. Line Emission and Escape

Since all level populations are calculated explicitly, line emissivities and cooling rates are calculated as a straightforward product of the population of the line upper level, the spontaneous transition probability and an escape fraction.

Line optical depths may be large in some nebular situations. Photons emitted near the centers of these lines are likely to be absorbed by the transition which emitted them and reemitted at a new frequency. This line scattering will repeat many times until the photon either escapes the gas, is destroyed by continuum photoabsorption or collisional deexcitation, or is degraded into longer wavelength photons which may then escape. Our treatment of resonance line transfer is based on the assumption of complete redistribution. That is, we assume that there is no correlation of photon frequencies before and after each scattering event. This has been shown to be a good approximation for a wide variety of situations, particularly when the line profile is dominated by Doppler broadening. In this case, more accurate numerical simulations (e.g., Hummer and Rybicki 1971) have shown that line scattering is restricted to a small spatial region near the point where the photons are emitted. Line photons first scatter to a frequency such that the gas cloud is optically thin and then escape in a single long flight. The probability of escape per scattering depends on the optical depth, τ_0 at the center of the line. For $1 \leq \tau_0 \leq 10^6$, the resonant trapping is effectively local. For $\tau_0 \geq 10^6$, the lines become optically thick in the damping wings, and the line escapes as a result of diffusion in both space and frequency. Since the scattering in the Doppler core is always dominated by complete redistribution, and since most of the lines in our models are optically thin in the wings, we assume that all line scattering takes place in the emission region.

We use the following expression for escape probability (Kwan and Krolik, 1981):

$$P_{esc.,line}(\tau_{line}) = \frac{1}{\tau_{line}\sqrt{\pi}(1.2+b)}(\tau_{line} \geq 1) \tag{6}$$

$$P_{esc.,line}(\tau_{line}) = \frac{1-e^{-2\tau_{line}}}{2\tau_{line}}(\tau_{line} \leq 1) \tag{7}$$

where

$$b = \frac{\sqrt{\log(\tau_{line})}}{1+\tau_{line}/\tau_w} \tag{8}$$

,

τ_{line} is the optical depth at line center, and $\tau_w = 10^5$.

The rates for line emission and the probabilities for the various resonance line escape and destruction probabilities depend on the state of the gas at each point in the cloud. The cooling function for the gas depends on the line escape probabilities, and the effects of line trapping must be incorporated in the solution for the temperature and ionization of the gas. Once the state of the gas at a given point has been determined, the emission in each line is calculated as the product of the upper level population and the corresponding net decay rate, including the suppression due to multiple scattering.

E.2. Continuum Emission

Diffuse continuum radiation is emitted by three processes: thermal bremsstrahlung, radiative recombination, and two-photon decays of metastable levels. The thermal bremsstrahlung emissivity is given by Osterbrock 1974:

$$j_\varepsilon = \frac{1}{4\pi} n_z n_e \frac{32 Z^2 e^4 h}{3 m^2 c^3} \left(\frac{\pi h \nu_0}{3kT} \right)^{1/2} e^{-h\nu/kT} g_{ff}(T, Z, r) \tag{9}$$

where T is the electron temperature, n_e is the electron abundance, Z is the charge on the most abundant ion, n_z is the abundance of that ion, and g_{ff} is a Gaunt factor (Karzas and Latter, 1966). For two photon decays, we adopt the distribution (Tucker and Koren 1971):

$$H\left(\frac{\varepsilon}{\varepsilon_0}\right) = 12 \left(\frac{\varepsilon}{\varepsilon_0}\right)^2 \left(1 - \frac{\varepsilon}{\varepsilon_0}\right) \tag{10}$$

where ε_0 is the excitation energy.

E.3. Continuum Transfer

The continuum radiation field is modified primarily by photoabsorption, for which the opacity, $\kappa(\varepsilon)$, is equal to the product of the ion abundance with the total photoionization cross section, summed over all levels.

A model is constructed by dividing the cloud into a set of concentric spherical shells. The radiation field incident on the innermost shell is the source spectrum. For each shell, starting with the innermost one, the ionization and temperature structure is calculated from the local balance equations using the radiation field incident on the inner surface. The attenuation of the incident radiation field by the shell is then calculated. The diffuse radiation emitted by the cloud is calculated using an expression of the formal solution if the equation of transfer:

$$L_\varepsilon = \int_{R_{inner}}^{R_{outer}} 4\pi R^2 j_\varepsilon(R) e^{-\tau_{cont.}(R,\varepsilon)} dR \tag{11}$$

where L_ε is the specific luminosity at the cloud boundary, $\tau_{cont.}(R,\varepsilon)$ is the optical depth from R to the boundary, and j_ε is the emissivity at the radius R. Since our models in general have two boundaries, we perform this calculation for radiation escaping at both the inner and outer cloud boundaries. This calculation is performed for each continuum energy bin, and separately for each line. In the case of the continuum, we construct a vector of emissivities, $j_\varepsilon(R)$ which includes contributions from the escaping fraction from all the levels which affect each energy. For the lines, the emissivity used in this equation is the escaping fraction for that line.

E.4. Radiation Field Quantities and Transfer Details

Equation (11) conceals a variety of important issues concerning the treatment of the radiation field and the values which are printed in the various output files produced by xstar. In an effort to clarify this we present here a complete description of the various radiation field quantities which are used internally to xstar, and which are output to the user. In this subsection, all radiation fields are specific luminosity, L_ε, in units erg/s/erg for the continuum, and luminosity, L_i, in units erg/s for lines. We distiguish several different radiation fields. First, the radiation field used locally by xstar for the calculation of photoionization rates and heating, we denote $L_\varepsilon^{(1)}$. This is calculated during an outward iteration using the transfer equation:

$$\frac{\mathrm{d}L_\varepsilon^{(1)}}{\mathrm{d}R} = -\kappa_{cont}(\varepsilon) L_\varepsilon^{(1)} + 4\pi R^2 j_\varepsilon(R) \tag{12}$$

with the boundary condition that $L_\varepsilon^{(1)} = L_\varepsilon^{(inc)}$ at the inner radius of the cloud. Here $\kappa_{cont}(\varepsilon)$ and j_ε are the local continuum opacity and emissivity and $L_\varepsilon^{(inc)}$ is the incident radiation field at the inner edge of the cloud.

In addition we can define the various radiation fields of interest for use in fitting to observed data. These include the spectrum transmitted by a model, i.e. the radiation which would be observed if the incident radiation field were subject to absorption alone:

$$L_\varepsilon^{(2)} = L_\varepsilon^{(inc)} e^{-\tau_{cont}^{(tot)}(\varepsilon)} \tag{13}$$

where $\tau_{cont}^{(tot)}(\varepsilon)$ is the total optical depth through the model cloud due to continuum photoabsorption,

$$\tau_{cont}^{(tot)}(\varepsilon) = \int_{R_{inner}}^{R_{outer}} \kappa_{cont}(\varepsilon) dR \tag{14}$$

Also of interest is the total emitted continuum radiation in both the inward and outward directions, which is given by equations similar to (11):

$$L_\varepsilon^{(3)} = \int_{R_{inner}}^{R_{outer}} 4\pi R^2 j_\varepsilon(R) e^{-\tau_{cont}^{(in)}(\varepsilon)} P_{esc,cont.}^{(in)}(R) dR \tag{15}$$

$$L_\varepsilon^{(4)} = \int_{R_{inner}}^{R_{outer}} 4\pi R^2 j_\varepsilon(R) e^{-\tau_{cont}^{(out)}(\varepsilon)} P_{esc,cont.}^{(out)}(R) dR \tag{16}$$

where the escape probabilities in the inward and outward directions are $P_{esc,cont.}^{(in)}(R) = (1 - C)/2$ and $P_{esc,cont.}^{(out)}(R) = (1 + C)/2$, where C is the covering fraction, specified as an input parameter, and $\tau_{cont}^{(out)}(\varepsilon)$ and $\tau_{cont}^{(out)}(\varepsilon)$ are the continuum optical depths in the inward and outward directions.

Line luminosities are calculated separately, one at a time, according to an equation analogous to equation (12):

$$\frac{dL_i^{(1)}}{dR} = -\kappa_{cont}(\varepsilon) L_i^{(1)} + 4\pi R^2 j_i(R) P_{esc,line}^{(in)} \tag{16}$$

$$\frac{dL_i^{(2)}}{dR} = -\kappa_{cont}(\varepsilon)L_i^{(2)} + 4\pi R^2 j_i(R) P_{esc,line}^{(out)} \tag{17}$$

where $L_i^{(1)}$ and $L_i^{(2)}$ are the luminosities of individual lines in the inward and outward directions, respectively. The escape probabilities in the inward and outward directions are calculated using $P_{esc.,line}(\tau_{line})$ from equations (6)-(8) and $P_{esc.,line}^{(in)} = (1-C)P_{esc.,line}(\tau_i^{(in)})$ and $P_{esc,line}^{(out)} = (1-C)P_{esc.,line}(\tau_i^{(out)}) + CP_{esc.,line}(\tau_i^{(out)} + \tau_i^{(in)})/2$, and $\tau_i^{(in)}$ and $\tau_i^{(out)}$ are the line scattering optical depths in the inward and outward directions:

$$\tau_i^{(in)}(R) = \int_{R_{inner}}^{R} \kappa_i \mathrm{dR} \tag{18}$$

$$\tau_i^{(out)}(R) = \int_{R}^{R_{outer}} \kappa_i \mathrm{dR} \tag{19}$$

.

and κ_i is the line center opacity.

None of the continuum luminosities defined in equations (12)-(16) have the effects of lines included, either in emission or absorption. This is because lines scatter the radiation, while photoionization is true absorption. The effects of lines on the continuum can be added to the continuum for the purposes of comparing with observed spectra by binning the lines, i.e. we can calculate the binned specific luminosity and opacity:

$$L_{line,\varepsilon}^{(in)} = \Sigma_{i \ni |\varepsilon_i - \varepsilon| \leq \Delta\varepsilon} \frac{L_i^{(1)} \phi(\varepsilon - \varepsilon_i)}{\Delta\varepsilon} \tag{20}$$

$$L_{line,\varepsilon}^{(out)} = \Sigma_{i \ni |\varepsilon_i - \varepsilon| \leq \Delta\varepsilon} \frac{L_i^{(2)} \phi(\varepsilon - \varepsilon_i)}{\Delta\varepsilon} \tag{21}$$

$$\kappa_{line}(\varepsilon) = \Sigma_{i \ni |\varepsilon_i - \varepsilon| \leq \Delta\varepsilon} \kappa_i \phi(\varepsilon - \varepsilon_i) \tag{22}$$

where ε and $\Delta\varepsilon$ are the energy and width, respectively, of the continuum bin closest to line i, and $\phi(\varepsilon - \varepsilon_i)$ is the profile function including the effects of broadening due to thermal Doppler motions, natural broadening, and turbulence.

Then we can define the total optical depth of the cloud

$$\tau^{(tot)}(\varepsilon) = \int_{R_{inner}}^{R_{outer}} (\kappa_{cont}(\varepsilon) + \kappa_{line}(\varepsilon))\mathrm{dR} \tag{23}$$

and the total transmitted specific luminosity

$$L_\varepsilon^{(5)} = L_\varepsilon^{(inc)} e^{-\tau^{(tot)}(\varepsilon)} \tag{24}$$

and the total emitted specific luminosity in the inward and outward directions:

$$L_\varepsilon^{(6)} = L_\varepsilon^{(3)} + L_{line,\varepsilon}^{(in)} \tag{25}$$

$$L_\varepsilon^{(7)} = L_\varepsilon^{(4)} + L_{line,\varepsilon}^{(out)} \tag{26}$$

The quantities $L_\varepsilon^{(inc)}$ $L_\varepsilon^{(5)}$, $L_\varepsilon^{(6)}$ and $L_\varepsilon^{(7)}$ are output in columns 2,3,4,5 of the file xout_spect1.fits. The quantities $L_\varepsilon^{(inc)}$ $L_\varepsilon^{(5)}$, $L_\varepsilon^{(3)}$ and $L_\varepsilon^{(4)}$ are output in columns 2,3,4,5 of the file xout_cont1.fits.

In fact, the lines should be included in the continuum which is responsible for the local ionization and heating of the gas, since they can contribute to these processes. So we define a modified version of equation (12):

$$\frac{\mathrm{d}L_\varepsilon^{(1')}}{\mathrm{d}R} = -\kappa_{cont}(\varepsilon)L_\varepsilon^{(1')} + 4\pi R^2 j_\varepsilon(R) + 4\pi R^2 \Sigma_{i\ni|\varepsilon_i-\varepsilon|\leq\Delta\varepsilon}\frac{j_i(R)P_{esc,line}^{(out)}\phi(\varepsilon-\varepsilon_i)}{\Delta\varepsilon} \quad (27)$$

$L_\varepsilon^{(1')}$ is the quantity which is used by xstar to calculate the local ionizing flux. This is the quantity which is conserved by xstar when it calculates heating=cooling.

The quantities $L_i^{(1)}$, $L_i^{(2)}$, $\tau_i^{(in)}$ and $\tau_i^{(out)}$ are output in columns 6,7,8,9 of the file xout_lines1.fits.

The quantities which contain the continuum only, before the lines are binned and added, are printed out to the log file, xout_step.log, when the print switch lpri is set to 1 or greater. Then they are in a table following the label 'continuum luminosities'. The quantities $L_\varepsilon^{(inc)}$, $L_\varepsilon^{(1')}$, $L_\varepsilon^{(3)}$, $L_\varepsilon^{(4)}$, $\tau_{cont}^{(in)}(\varepsilon)$ and $\tau_{cont}^{(out)}(\varepsilon)$ are output in columns 3,4,5,6,7 and 8. Many other useful quantities are output to the log file when lpri=1. This includes the quantities $L_i^{(1)}$, $L_i^{(2)}$, $\tau_i^{(in)}$ and $\tau_i^{(out)}$ are in columns 2-5 following the label 'line luminosities'

E.5. Energy Conservation

Energy conservation is imposed as a constraint when determining the temperature in xstar when the input parameter niter is non-zero. If so, the temperature is iteratively improved until the heating and cooling rates are locally equal. This is implemented by calculating the integral over the absorbed and emitted continuum energy in a given spatial zone, and also the sum over the energy emitted in the lines. Compton heating and cooling are added analytically, since Comptonization of the radiation field is not treated. The error resulting from this procedure is tabulated in the log file 'xout_step.log' in the 8th column of the step-by-step output, in units of %.

Energy conservation locally should correspond to global energy conservation, i.e. that the total absorbed energy in the radiation field equals the total emitted energy in lines plus continuum. This is tested at each spatial zone in xstar by calculating

$\int (L_\varepsilon^{(inc)} - L_\varepsilon^{(1)})d\varepsilon - \Sigma_i(L_i^{(1)} + L_i^{(2)})$. The error resulting from this procedure is tabulated in the log file 'xout_step.log' in the 9th column of the step-by-step output, in units of %.

It is important to point out that the specific luminosities in the file 'xout_spect1.fits' 'xout_cont1.fits' are not expected, in general, to show energy conservation. This is primarily because the transmitted spectra in both of these files contain the effects of binned lines. Line opacity is expected to produce a scattering event, i.e. the photon is likely to be reemitted near the same energy. This differs qualitatively from photoelectric absorption, in which an absorbed photon is likely to be reemitted at a very different energy, with an accompanying net loss or gain of energy to the electron thermal bath. Line opacity is not included in the radiative equilibrium integral used to calculate the gas temperature, and so the total absorbed energy in the radiation field $L_\varepsilon^{(5)}$ will in general not equal the emitted energy in $L_\varepsilon^{(6)} + L_\varepsilon^{(7)}$. Energy conservation can be checked using the quantities $L_\varepsilon^{(1)}$, $L_i^{(1)}$ and $L_i^{(2)}$ from the file xout_step.log.

Energy conservation checked using binned line spectra is also affected by the errors introduced by binning. This is discussed at length in the section of this manual on table models for xspec, but we emphasize here that the binned spectrum cannot be accurately integrated to derive the total line absorption or emission unless the lines are broad compared with the energy grid spacing (requiring turbulent velocities ≥ 500 km/s currently).

E.6. Algorithm

Construction of a model of an X-ray illuminated cloud consists of the simultaneous solution of the local balance equations. The radiative transfer equation is solved for both the continuum and for the lines that escape the region near the point of emission. The large number of ions in the calculation results in many ionization edges that may affect the radiation field. We solve the transfer equation on a frequency grid that includes a total of 9999 continuum grid points with even logarithmic spacing in energy from 0.1 eV to 20 keV resulting in a limiting resolution of 0.12 %, corresponding to, e.g. 8.6 eV at 7 keV. We calculate the luminosities of ∼10000 spectral lines and solve the continuum transfer equation individually for each of these. The emissivity of each line at each point is the product of the emissivity and the local escape fraction for that line. The continuum opacity for each line is the opacity calculated for the energy bin that contains the line. This procedure is repeated for each successive shell with increasing radius.

Calculation of the escape of the diffuse radiation field depends on a knowledge of the optical depths of the cloud from any point to both the inner and outer boundaries. Since these are not known a priori we iteratively calculate the cloud structure by stepping through the radial shells at least 3 times. For the initial pass through the shells we assume that the optical depths in the outward direction are zero. This procedure is found to converge satisfactorily within 3-5 passes for most problems of interest. This procedure is tantamount to the "Λ-iteration" procedure familiar from stellar atmospheres, and must suffer from the same convergence problems when applied to problems with large optical depths. These problems are reduced in our case by the use of escape probabilities rather than a full integration of the equation of transfer.

F. Atomic Processes

Here we summarize the most important data sources adopted for the calculations. These are discussed in greater detail, along with a description of the fitting formulas and assumptions, in Bautista and Kallman 2000.

F.1. Photoionization

Photoionization rates are obtained by convolving the radiation field with the photoionization cross section. Cross sections are included for all levels of every ion for a wide range of photon energies occurring in our model. The cross sections where taken from the Opacity Project (Seaton, et al., 1993, Cunto et al., 1993), then averaged over resonances as in Bautista et al. 1997 and split over fine structure according to statistical weights.

For inner shell photoionization not yet available from the opacity project we use the cross sections of Verner and Yakovlev 1995. Inner shell ionization in X-ray illuminated clouds is enhanced by Auger cascades. This process can result in the ejection of up to eight extra electrons (in the case of iron) in addition to the original photoelectron. We include this effect by treating each inner shell ionization/auger event as a rate connecting the ground state of one ion with another level of an ion (in general not adjacent to the initial ion). The rates for inner shell ionization/auger processes are calculated using the relative probabilities of the various possible outcomes of an inner shell ionization event from Kaastra and Mewe 1989. These yields are multiplied by the appropriate inner shell photoionization cross section in order to calculate a rate for each inner shell ionization/Auger cascade individually. Our level scheme also includes levels with inner shell vacancies, which are populated by inner shell/Auger events. Populations of these levels are calculated in the customary way, and the decays from these levels produce inner shell flourescence lines.

F.2. Collisional Ionization

Ionization by electron collisions is important if the gas temperature approaches a fraction of the ionization threshold energy of the most abundant ions in the gas. For ground states we include the rates from Raymond and Smith 1986 for elements other than iron, and from Arnaud and Raymond 1992 for iron. Collisional ionization from excited levels may also be important to the ionization balance. We include ionization rates for all excited levels of every ion using approximate formulae by D. Sampson and coworkers (Zhang and Sampson, 1987). 3-body recombination rates to all levels are calculated from the collisional ionization rates using the detailed balance principle.

F.3. Recombination

Radiative and dielectronic recombination rates to all spectroscopic levels are calculated from the photoionization cross sections using the Milne relation. We include both spontaneous and stimulated recombination caused by the illuminating radiation. Stimulated recombination by the locally emitted radiation is not treated explicitly, although its effect is taken into account in an approximate way by suppressing a fraction of the spontaneous recombinations using the escape probability described earlier in this section. Recombination onto the superlevels is calculated in order to account for the difference between the sum over all spectroscopic levels and the total ion recombination as given by Nahar and coworkers (Nahar 1999, Nahar 2000) where available and Aldrovandi and Pequignot 1973 for species other than iron ions and ions in the H and He isoelectronic sequences. For iron we use total rates from Arnaud and Raymond 1992. For H and He-like ions the total recombination rates were calculated by Bautista et al. 1998 and Bautista and Kallman 2000.

F.4. Collisional Excitation and Radiative Transition Probabilities

Collision strengths and A-values were collected from a large number of sources. Particularly important for this compilation were the CHIANTI data base (Dere, et al., 1997) for X-ray and EUV lines, and the extensive R-matrix calculations by the Iron Project (Hummer et al., 1993).

F.5. Charge Transfer

Rates for charge transfer reactions are taken from Butler Heil and Dalgarno 1980. For highly charged ions, where accurate calculations do not exist, we scale the rates along isonuclear sequences, assuming that the cross section is proportional to the square of the total residual charge transfer reaction of O II with H (Field and Steigman 1971).

Chapter 10

Theory of Operation

A. XSTAR

A.1. Introduction

Although XSTAR is designed with the goal of maximizing the flexibility of parameter values and assumptions available to the user, there are likely to arise situations in which the standard set of input options are not sufficient. Under these circumstances the user may want to attempt to modify the source code in order to effect a particular set of assumptions, geometry, etc. Problems for which customization is more likely to be necessary include models with additional heating or cooling processes (for example adiabatic expansion cooling or cosmic ray heating) or different assumptions about line escape probability (e.g. Sobolev escape probability in a medium with a velocity gradient). If so, the internal operation of the code must be confronted. This appendix presents a summary of the code operation in order to aid in code customization.

A.2. Programming Philosophy

XSTAR has been written and developed over a span of approximately 15 years, and much has changed during that time. Changes include advances in the fortran language, changes in the speed and memory capacity of the available computers, new insights into the flexibility required of XSTAR, and insight into which physical processes are likely to have the greatest effect on the model results.

XSTAR was written in standard fortran 77, and much of the code retains the programming style associated with the older versions of the fortran language. Hard experience with moving the code from one type of machine to another has led to an attempt to avoid any machine dependent extensions to the fortran language, any word length dependent numerical constructs or any use of extended precision arithmetic (with one or two exceptions).

The code is structured in an attempt to be modular, and to separate the calculation of the atomic rates from the calculation of level populations, ion fractions, temperature, etc. The goal is to make it relatively easy to add or change atomic data, requiring modification of just one subroutine if a new process is added or a fitting formula is changed. The data itself is all read in from an external file, so that it can be changed without any modification to the code itself if the existing fitting formulas are unchanged.

The remainder of this appendix is organized as follows. First we present a list of the subroutines and a short description of their function. This list is sorted by category: subroutines used in input, output, defining ionizing spectrum, primary driving subroutines, and rate calculation. Then we present a list of variable names and a description of their contents. Finally we present a schematic flow chart of the operation of XSTAR.

A.3. List of Subroutines

Primary Computational Subroutines

xstar: Main program.

cndnse: condense matrix for level population calculation

dsec: calculate thermal equilibrium and charge neutrality using secant method

ener: set up energy grid

func: calculate all ion fractions, level populations, heating, cooling.

func1: calculate rates affecting ion fractions

func2: calculate rates affecting level populations

func2a: calculate rates affecting level populations due to Auger and inner shell photoionization

func2i: calculate number of bound levels for an ion

func3: calculate heating-cooling rates

init: initialization. Zeroes most variables

heatt: total heating and cooling rate calculation

hcor1: Level population calculation

ioneqm: calculation ion fractions

istruc: calculate ion abundances

invert: used to prepare for next iteration of global calculation

step: calculates spatial step size

stpcut: update important quantities after each spatial step.

trnfrc: calculates dilution and attenuation of incident radiation

trnfrn: calculates transfer of diffusely emitted radiation.

Rate Calculation Subroutines

amcol: calculate rate of collisional angular momentum changing transitions in hydrogenic ions using method of Pengelly & Seaton (1964).

amcrs: calculate rate of collisional angular momentum changing transition in hydrogenic ions using either **amcol** or **impact**.

anl1: calculate allowed radiative transition rates for hydrogenic ions.

bkhsgo:

calt60_62: returns Callaway Upsilons for H-like ions.

calt66: returns Kato & Nakazaki (1989) Upsilons for He-like ions.

calt67: returns Keenan et al.(1987) Upsilons for He-like ions.

calt68: returns Sampson & Zhang Upsilons for He-like ions.

calt69: returns Kato & Nakazaki (1989) Upsilons for He-like ions.

calt70: takes type 70 data and calculates recombination rates and phot. cross sections of superlevels.

calt71: returns radiative trans. rates from superlevels to spectroscopic levels.

calt72: returns capture rates for DR through satellite levels.

calt73: returns Upsilons for satellite levels of He-like ions.

calt74: returns levels specific DR and calculates photoionization rates over DR associated resonances.

calt77: returns collisional rates between superlevels and spectroscopic levels.

erc: calculate collisional excitation rates for hydrogenic ions.

hgf: calculates hypergeometric functions.

impact: impact parameter cross sections.

impactn: calculate collisional excitation rates for hydrogenic ions using the impact parameter.

impcfn: calculate the functions used in the impact parameter.

intin:

intin2:

intino:

irc: calculate collisional ionization rates using the Johnson (1972) method for hydrogenic ions or routine **szirc** for other ions.

levwk: calculates partition functions

milne: calculates milne integral

phintf: photoionization rate and milne rate integrator.

szcoll: calculates electron impact excitation rates from semiempirical formula (eq.35) from Sampson & Zhang (1988, apj 335, 516)

szirc: calculates electron impact ionization rates from semiempirical formula (eq.35) from Sampson & hang (1988, apj 335, 516)

ucalc: main rate calculation routine

upsil: returns Upsilons from Burgess & Tully fits.

velimp: calculate collisional excitation rate from the impact parameter.

Input Subroutines

readtabl1: reads atomic pointer fits file

readtabl2: reads atomic data fits file

rread1: reads in commands. calls xpi routines

unsavd: the opposite of savd

Output Subroutines

bnchmrk: comparison with Lexington benchmarks

deletefile: deletes unneeded fits files

fheader: writes fits header

fitsclose: close fits files

fparmlist: writes fits parameter list

fstepr:

fwrtascii: writes ascii fits files

savd: Saves temporary data after each spatial zone for iterative calculation

pprint: main printout routine.

printerror: Prints errors encountered during fits i/o.

writespectra: writes out fits file with spectrum at end of calculation.

Atomic Database Subroutines

dbwk: main database manipulation routine. sets up atomic database pointers among other things

dprint: write out a record to the database

dprinto: write out a record in ascii

dprints: write a one line summary of record in ascii

dread: read a record

dreado: read a record in ascii

Miscellaneous Subroutines

bremem: calculate bremsstrahlung emissivity

comp: calculate non-relativistic Compton heating and cooling

dexpo: real*8 expo

dfact: real*8 fact

ee1exp: first exponential integral*exponential

ee1expo: first exponential integral*exponential

eint: first exponential integral

enxt: find next energy for trapezoid quadrature

expo: exponential function with boundaries

exint1:

exp10: 10^x with boundaries

expint: first exponential integral*exponential

fact: factorial

fact8: real*8 factorial

fbg: used in calculating bremsstrahlung emissivity (Raymond and Smith 1986)

hlike: hydrogenic photoionization cross section

hunt: table search (Numerical Recipes)

huntf: table search assuming logarithmic spacing

ispec: input spectrum from thermal brems with unit gaunt factor

ispcg2:

ispec4: input spectrum, single power law

ispecg: input spectrum, from atable

ispecgg: input spectrum, generic renormalization

leqt2f: solves linear system

lubksb: used in linear system solution. from Numerical Recipes.

ludcmp: used in linear system solution. from Numerical Recipes.

mprove: used in linear system solution. from Numerical Recipes.

nbinc: finds continuum bin for given energy

remtms: fake ibm routine to calculate remaining cpu time in msec.

spline: Spline fit

splinem: Spline fit

splint: Spline fit

pescl: calculates line escape probability

pescv: calculates continuum escape probability

starf: calculates blackbody spectrum

A.4. Units

An attempt has been made to retain the same units for a given physical quantity throughout the code.

Temperatures: 10^4 K

Distances: cm

Total Luminosities: 10^{38} erg/s

Photon energies: eV

Continuum Emissivities (Specific): erg cm^{-3}s^{-1} erg^{-1}

Continuum Luminosities (Specific): 10^{38} erg s^{-1} erg^{-1}

Line Emissivities: erg cm^{-3} s^{-1}

Line Luminosities: 10^{38} erg s^{-1}

Heating and Cooling rates: erg cm^{-3} s^{-1}

Photoionization Rate Coefficients: s^{-1}

Collisional Rate Coefficients: cm^3 s^{-1}

A.5. List of Variable Names

tp is the radiation temperature, in units of KeV

xlum is the x-ray source luminosity, in units of 10**38 erg/s

ecut is the low energy cutoff of the input spectrum

lpri is the print switch ,1=on, 0=off.

lwri is the write switch.

nel is the number of elements (max=13)

nnnl is the maximum number of lines (1530)

nni is the number of ions, excluding fully stripped (max=168)

ktitle is the model title

r is the radius (cm)

delr =r-rl

rstp =delr/r

rdel is the distance from the current r to the first r of the calculation (cm)

t is the temperature (10**4 k)

xee : electron relative abundance,

xpx : total particle density (/cm**3)

xnx : electron number density (/cm**3).

abel : the relative element abundances.

xii : the relative ion abundances.

epi : the continuum energy bins and the difference between

bremsa : the local ionizing flux (erg/erg/cm**2/sec)

zrems : the specific luminosity. (10**38 erg/erg/sec).

zeta : ionization parameter log(l/(n*r**2)) (

opakc : continuum opacities. units are /cm.

tau0 : line center optical depths from the cloud center.

B. Flow Chart

C. XSTAR2XSPEC

XSTAR2XSPEC consists of three major components (in addition to XSTAR itself).

xstar2xspec: xstar2xspec is a Perl script which manages the overall program flow of XSTAR2XSPEC.

xstinitable: xstinitable is an FTOOL used in the initialization phase of XSTAR2XSPEC. It builds the FITS PARAMETER table from the input data (xstinitable.fits) and a text file (xstinitable.lis) which is basically of list of all the calls to XSTAR to generate the required spectra.

xstar2table: xstar2table is an FTOOL called after xstar in each iteration of the xstar2xspec loop.

C.1. xstar2xspec (script)

This is a Perl script.

One current limitation is that the file tagging in the `-save` option file names are limited to 9999 calls of XSTAR.

C.2. xstinitable

This FTOOL is written in C. It generates an initial FITS file (xstinitable.fits) with the appropriate PRIMARY and PARAMETERS extension from the atables and mtable file are build. It also generates a text file (xstinitable.lis) which contains a complete XSTAR calling command on each line. This file is processed by the XSTAR2XSPEC Perl script

XSTAR

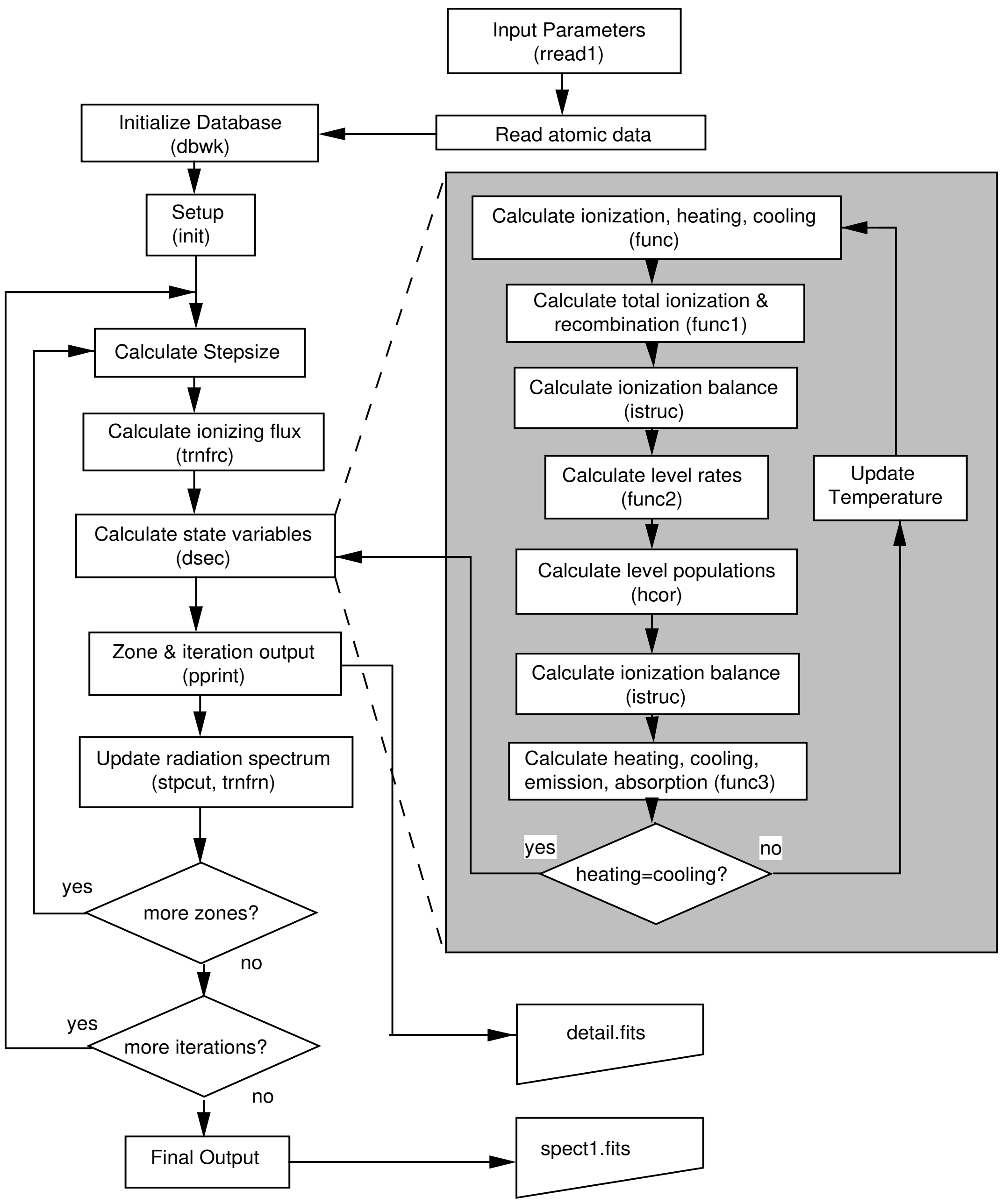

Fig. 1.— The basic program flow of XSTAR.

XSTAR2XSPEC

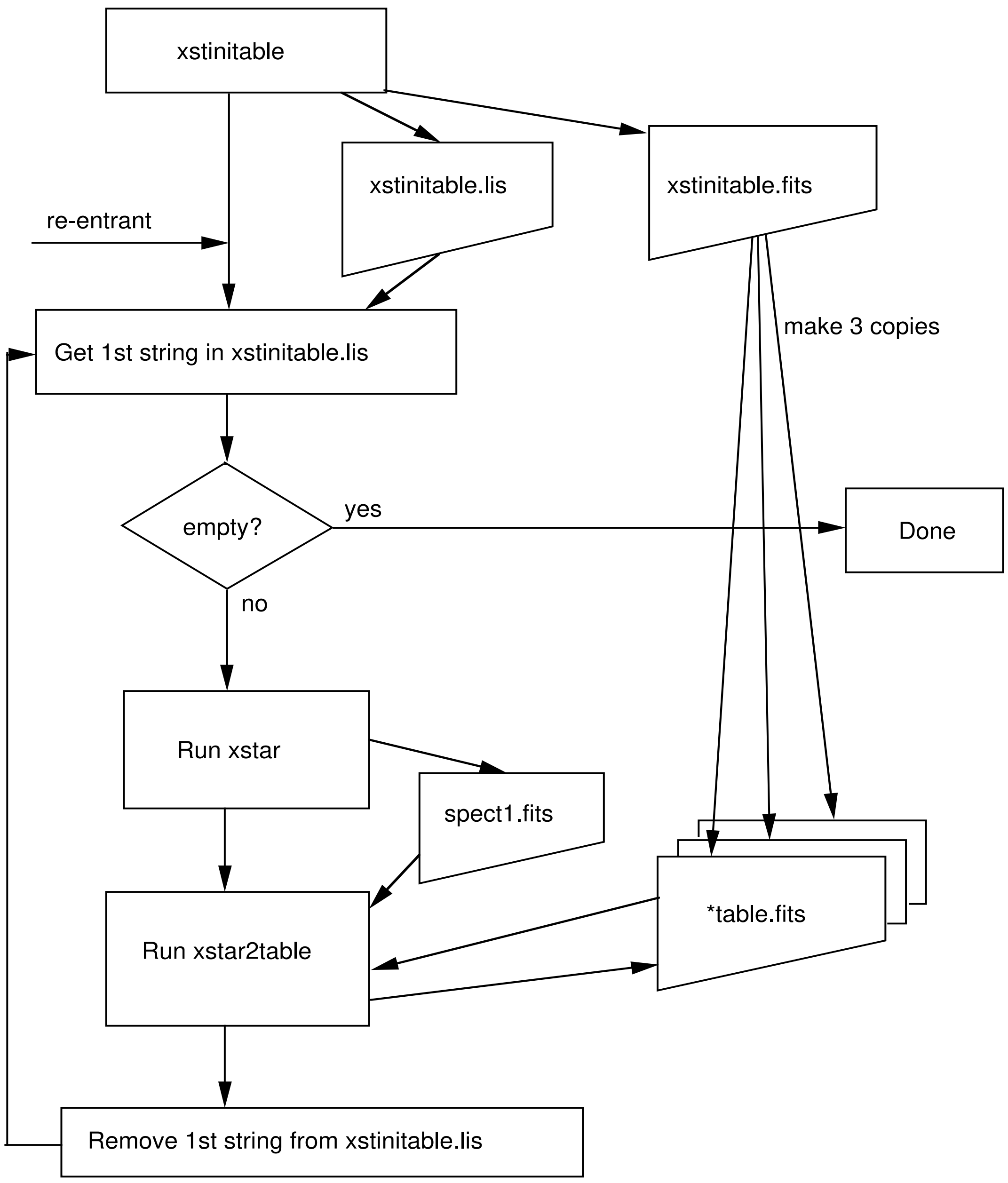

Fig. 2.— The basic program flow of XSTAR2XSPEC.

Examples of changes in XSTAR that would require changes in this FTOOL (just meant as a sample, not necessarily an exhaustive list).

1. Changing the number of physical parameters in XSTAR.
2. Changing or altering the control parameters in XSTAR.

C.3. xstar2table

This FTOOL is also written in C. Full use is made of dynamic memory allocation. If the number of spectral channels is changed in XSTAR, this program should adapt appropriately, automatically.

Examples of changes in XSTAR that would require changes in this FTOOL (just meant as a sample, not necessarily an exhaustive list).

1. Changing the number of physical parameters in XSTAR.
2. At present, this program is written assuming that the XSTAR runs are performed sequentially and in a particular order. In fact, it checks the sequence by comparing the LASTSPEC keyword to the LOOPCONTROL variable. If, at some time in the future, the XPI interface is modified in such a way that it becomes possible to simultaneously submit multiple XSTAR runs on multiple processors, xstar2table must be modified to ignore this comparison.

Chapter 11

The Lexington Benchmarks

A. Introduction

In May, 1994, the Lexington conference on photoionization modeling was held in order to better understand the differences between the various codes available for solving photoionization problems. The conference was attended by approximately 8 modelers actively involved in the solution of problems involving H II regions, nova envelopes, planetary nebulae, and AGN clouds. As part of the workshop the participants were asked to run several standard problems and compare the results. These were chosen from what were thought to be representative models of H II regions, planetary nebulae, and AGN clouds. The results will appear in a book on the proceedings of the Space Telescope Institute Conference on Emission Lines. In this appendix we reproduce the comparison with the results of XSTAR 1.17 with the other models given in the conference proceedings, along with the input files needed to reproduce these results. Note that version 1.17 has added an input command 'benchmark n', where n is an integer from 1 to 8, representing the test case in the Lexington paper. This command produces a table, similar to the 'print line_fluxes' command, but also including the mean value and dispersion of each line flux from the ensemble of model results in the Lexington paper (with the results of the current run substituted for the XSTAR results given there; columns 5 and 6), the flux from the current XSTAR run (in the appropriate units asadopted by the Lexington conference; column 7), and the number of 'sigmas' the XSTAR result differs from the mean (column 4). The last line gives the summed χ^2 for all lines from the 10 codes represented in the Lexington paper. The first two entries are always zero; the XSTAR χ^2 is in the 4th non-zero column (column 6).

B. Cool H II Region (benchmark 1)

B.1. Input:

C. Meudon H II Region (benchmark 2)

C.1. Input:

C.2. Output:

benchmark number 2

1 He I 5876.00 1.24 1.17E-01 7.96E-03 1.27E-01

2 Cll 2326.00 0.51 1.64E-01 5.20E-02 1.91E-01

3 C 1909.00 1.59 7.06E-02 1.15E-02 8.88E-02

4 [N II] 1220000.00-0.59 3.31E-02 2.28E-03 3.18E-02

5 [N II] 6584.00 0.67 7.78E-01 7.43E-02 8.28E-01

6 [N III] 570000.00-1.29 2.58E-01 6.89E-02 1.69E-01

7 [0 II] 3727.00 2.04 2.19E+00 3.23E-01 2.85E+00

8 [0 III] 518000.00-0.24 1.07E+00 6.84E-02 1.05E+00

9 [0 III] 884000.00-0.06 1.23E+00 9.18E-02 1.23E+00

10 [0 III] 5007.00 1.11 2.14E+00 1.52E-01 2.31E+00

11 [N III] 128000.00-0.72 2.14E-01 1.96E-02 2.00E-01

12 [Ne III] 155000.00 1.96 3.93E-01 7.86E-02 5.47E-01

13 [Ne III] 3869.00 2.80 9.41E-02 2.18E-02 1.55E-01

14 [S II] 6720.00 2.49 2.09E-01 6.30E-02 3.66E-01

15 [S III] 187000.00-0.69 5.46E-01 3.30E-02 5.24E-01

16 [S III] 340000.00-0.98 8.84E-01 4.30E-02 8.42E-01

17 [S III] 9532.00-0.29 1.31E+00 9.19E-02 1.28E+00

18 [SIV] 105000.00 0.21 3.37E-01 6.33E-02 3.50E-01

24 H I 4862.00-0.23 2.05E+00 4.23E-02 2.04E+00

0. 0. 38.1802 17.4427 8.74325 32.2630 18.6087 6.13845 7.23243 30.8659 10.6705 16.8548 32.2630

D. Blister H II Region (benchmark 3)

D.1. Input:

D.2. Output:

Chapter 12

Troubleshooting

A. XSTAR

Xstar will generate error messages when some of the input paramter files are outside the recommended range. The ftools i/o routines will not allow parameter values outside the prescribed limits, and will halt with very unhelpful i/o errors, but these limits can be changed by editing the parameter file.

B. XSTAR2XSPEC

C. When The Above Doesn't Work

If the above does not help in resolving the problem, you should contact Tim Kallman (address below). Be sure to include a copy of the XSTAR or XSTAR2XSPEC log file and the uname value of the machine you are running it on.

Timothy Kallman, Code 662	Internet:	tim@xstar.gsfc.nasa.gov
NASA GSFC	Telephone:	(301)-286-3680
Greenbelt MD 20771, USA	FAX:	(301)-286-1682

Chapter 13

The Atomic Database

A. General Description

The database system used by XSTAR version 2 attempts to separate, as much as possible, the numerical quantities which determine the various atomic rates from the fortran code which actually performs the calculation. The goal is make the atomic data modular, so that new data can be adopted or tested without requiring extensive modifications to the code. The way this is done is to separate the data from the code itself, and store the data in a database which is designed specifically for use by XSTAR. The database is divided into 'records', each of which corresponds to a given physical process affecting a given level or pair of levels. An example is the radiative decay of hydrogen from the 2p to the 1s level. Each record contains numerical constants needed to calculate the rate for the process, in this example simply the Einstein A value for the transition, together with various other associated quantities. Chief among these are two integers which describe the how the constants are to be used. The first integer is denoted the 'data type', and describes the fitting formula to be used in order to calculate a rate from the constants. The second integer is the 'rate type', which describes how XSTAR uses the rates calculated. The list of data types is already quite long and is expected to grow and change as new data is adopted into the database, but not all data types are used by the current database. In order to interpret the various data types, XSTAR contains one central data calculating subroutine, denoted ucalc.f, which branches to various segments of code (and calls to specialized subroutines) which are tailored to each data type. ucalc.f returns the rates in a standard form for use by the other XSTAR subroutines. It is expected that ucalc.f will require additions in order to handle new data types as they are adopted. The list of rate types is not intended to grow, since such changes could require changes to the rest of the XSTAR code structure.

The XSTAR database system can be divided into 3 parts:

First, and most important, is the ascii file containing all the data. That is, this file contains all the numerical data and labels required for calculation of all atomic rates and resultant quantities. This includes all level excitation energies, statistical weights and spectroscopic names, all element names and abundances, all ion names, and of course all photoionization cross sections, collision rates, recombination rates, fluorescence yeilds, and line wavelengths. This file is separated into records, corresponding crudely to lines of text, although many records extend over more than one line. Each record consists of a header, followed by the data. The header currently consists of 6 integers: the data type, the rate type, a continuation flag (currently unused), the number of reals in the record, the number of integers in the record, and the numbers of characters in the record. Then follows the real data, the integer data, and the character data. The various fields within the record are separated by one or more spaces. The record is terminated with a %, and the entire database is terminated by a single line containing %%%%. Each record can currently contain up to 2000 of any of the types of constants: real, integer, or character. In the XSTAR source tree this file is named atdat.text and currently is approximately 10MB in size.

In order to facilitate rapid reading of this file by XSTAR, it is converted into two binary fits tables. The first one contains the header data for each record, the second contains a concatenation of all the non-header data. They are named, respectively, aptrs.fits and atdat.fits. Operation of XSTAR requires that the environment variable LHEA_DATA be set to the directory containing these files. In the standard distribution these data files are kept in the refdata directory in the ftools area, and the appropriate value of the environment variable LHEA_DATA is set by the script which initializes FTOOLS, lhea-init.csh (c shell) or lhea-init.sh (bourne shell).

The third part of the database is the subroutine ucalc.f. This routine, when passed the contents of a record, returns the result of the rate calculation for the corresponding process. ucalc therefore contains all of the various arithmetic expressions corresponding to rates for various physical processes. ucalc returns generally 4 real rates and two integers. The rates are: rate, inverse rate, heating rate, and cooling rate. The integers are indeces of the levels involved, lower and upper. Not all data types return all 4 rates.

The list of rate types currently included in ucalc.f are as follows:

1 ground state ionization

3 bound-bound collision

4 bound-bound radiative

5 bound-free collision (level)

6 total recombination

7 bound-free radiative (level)

8 total recombination, forces norm

9 2 photon decay

11 element data

12 ion data

13 level data

14 bound-bound radiative superlevel-spectroscopic level

40 bound-bound collisional superlevel-spectroscopic level

The list of data types currently included in ucalc.f are as follows: Those denoted xstar1 are not in use in the standard distribution of XSTAR version 2, but are maintained in order to facilitate comparison with the results of XSTAR version 1.

In what follows, T=temperature in 10^4K; r1,r2,...=real numbers in record; i1,i2,...=integers in record; c1,c2,...=characters in record

1) Radiative recombination, Aldrovandi and Pequignot formula:

$$rate = r1/T^{r2}$$

2) Charge exchange with H^0, from Kingdon and Ferland: `rate=aax*expo(log(t)*bbx)*(1.+ccx*expo(ddx*t))*(1.e-9)`

```
ans1=rate*xh0
```

where xh0 is the number density of neutral hydrogen.

3) Autoionization correction to DR, formula from Hamilton, Sarazin and Chevalier:(xstar1)

$$rate = r1 * e^{r2/kT}/T^{1/2}$$

4) Line data: r1=line wavelength (A) ; r2=f value ; i1=lower level index ; i2=upper level index

5) 2 photon transition, collisional excitation.(xstar1) r1=line wavelength (A) ; r2=f value ; r3=lower level g ; r4=upper level g ; i1=lower level index ; i2=upper level index ; r5=collision strength at kT= ε_{line}

6) Level data: Contains level information for ions. Energies are in eV. Every line contains: reals r1=E_Z=1(eV); r2=(2J+1); r3=$n_{effective}$; r4=Ion. Potential; i1=n; i2=(2S+1); i3=L; i4=Z; i5=#lev; i6=#ion) ; c1-c#=configuration.

7) Dielectronic recombination, aldrovandi and Pequignot formula:

$$rate = r1 * 10^{-6} * e^{-r2/T} * (1 + r3 * e^{-r4/T})/T^{3/2}$$

8) Dielectronic recombination, Arnaud and Raymond formula:

$$rate = r1 * e^{-r2/kT} + r3 * (T^{-r4-r5*ln(T)})$$

9) Charge exchange with He^0, formula from Kingdon and Ferland:

10) Charge exchange with H^+, formula from Kingdon and Ferland: :]

```
 rate=aax*t**bbx*(1.+ccx*expo(ddx*t))
$             *expo(-eex/t)*(1.e-9)
 ans1=rate*xh1
```

11) 2 photon decay data:(xstar1) ; r1=line wavelength (A) ; r2=f-value ; i1=lower level index ; i2=upper level index

12) Photoionization cross section, broken power law: (xstar1)

$$\sigma = r1 * (\varepsilon/r5)^{r2}$$

i1=level index

13) element data: r1=abundance, r2=mass, i1=int(z), i2=index, c1-c30=name

14) Ion data: r1=ionization threshold, c1-c8=name

15) Photoionization cross section, Barfield, Koontz & Huebner scaled from neutrals: (xstar1)

$$crosssection = 10^{\Sigma_{n=1}^{n=12} C_n (ln(\varepsilon/\varepsilon_0))^{n-1} - 18}$$

for $\varepsilon \leq$ r3 ; where ε=photon energy in eV,ε_0=r1-Δ, Δ=r2, and C_1=r4, C_2=r5.

17) Line collision data: hydrogenic isosequence, rates from Cota:(xstar1)

r1=line wavelength (A) ; r2=f value ; r3=lower level g ; r4=upper level g ; i1=lower level index ; i2=upper level index ; r5=collision strength at kT= ε_{line}

18) Radiative Recombination rates for H-like levels, rates from Cota:(xstar1)

$$rate = 10^{r1+r2*(log_{10}(T)+4.-r3)^2}$$

19) Photoionization cross section from HULLAC: (not used).

20) Same as 10, but used for total rate

21)

22) Dielectronic Recombination, Storey low temperature: (xstar1)

$$rate = 10^{-12} * (r1/T + r2 + T * (r3 + T * r4)) * T^{3/2} * e^{-r5/T}$$

23) Photoionization cross section, Clark et al. formula: (xstar1)

25) Collisional Ionization data from Raymond and Smith:(xstar1) r1=e ; r2=a ; r3=b ; r4=c ; r5=d

$$ch = 1./chi$$

$$fchi = 0.3 * ch * (a + b * (1. + ch) + (c - (a + b * (2. + ch)) * ch) * alpha + d * beta * ch)$$

$$rate = 2.2e - 6 * sqrt(chir) * fchi * expo(-1./chir)/(e * sqrt(e))$$

26) Collisional Ionization data from Cota, H-like levels:(xstar1)

idest1=idat(1) ; gglo=rlev(2,idest1) ; edelt=abs(rlev(1,idest1)-rlev(1,nlev)) ; ans1=(4.1416e-9)*rdat(1)*t**rdat(2)*exp(-edelt/ekt) /gglo ; ans1=ans1*xnx

27) photoionization: hydrogenic

28) line data collisional: Mendoza; Raymond and Smith: (xstar1)

29) collisional ionization data: scaled hydrogenic: (xstar1)

30) Radiative Recombination, hydrogenic, total, Gould and Thakur formula:

$$rate = 2 * (2.105 \times 10^{-22}) * vth * y * \phi$$

where:

```
        zeff=r1
        beta=zeff*zeff/(6.34*t6)
        yy=beta
        vth=(3.10782e+7)*sqrt(t)
c       fudge factor makes the 2 expressions join smoothly
        ypow=min(1.,(0.06376)/yy/yy)
        fudge=1.*(1.-ypow)+(1./1.5)*ypow
        phi1=(1.735+alog(yy)+1./6./yy)*fudge/2.
        phi2=yy*(-1.202*alog(yy)-0.298)
        phi=phi1
        if (yy.lt.0.2525) phi=phi2
```

31) line data including statistical weights for upper and lower(xstar1)

32) Collisional ionization, Cota, ground level.(xstar1) idest1=idat(1)=1 ; gglo=rlev(2,idest1) ; edelt=abs(rlev(1,idest1)-rlev(1,nlev)) ; ans1=(4.1416e-9)*rdat(1)*t**rdat(2)*exp(-edelt/ekt) /gglo ; ans1=ans1*xnx

33) line data collisional: hullac (not used)

34) line data radiative: Mendoza and from Raymond and Smith (xstar1)

35) photoionization: table (from Barfield, Koontz and Huebner 1972): (xstar1)

36) photoionization, excited levels, hydrogenic(no l): (xstar1) i1=n, i4=level; i5=ion

49) opacity project pi x-sections Photoionization cross section from TOPbase averaged over resonances. Photon energies are in Ry with respect to the subshell ionization threshold and cross sections are in Mb. Just like 53, except for energy scale.. Every line contains: reals (2*np; x(i),y(i),i=1,np) ; integers (8; N, L, 2*J, Z, #lev+, #nion+, #nlev, #nion) ; characters (0) ; (#ion & #nlev correspond to the initial state and #ion+ & #lev+ correspond to the state to which that ionizes.)

50) line rad. rates from OP and IP Transition probabilities file. Every line contains: reals (3; Wavelength (A), gf-val., A-val. (cm-1)) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

51) iron project and chianti line collision rates Burgess & Tully fit to collision strengths as taken from CHIANTI. Each fit entry includes the C-fitting parameter and 5 reduced collision strengths values for X=0, .25, .5, .75, 1. ; reals (7; Delta E, C fitting param., 5 Y-reduced values) ; integers (5; transition type, lower level, upper level, Z, #ion) ; characters (0)

52) same as 59 but rate type 7

53) opacity project pi x-sections Photoionization cross section from TOPbase averaged over resonances. Photon energies are in Ry with respect to the first ionization threshold and cross sections are in Mb. Every line contains: reals (2*np; x(i),y(i),i=1,np) ; integers (8; N, L, 2*J, Z, #lev+, #nion+, #nlev, #nion) ; characters (0) ; (#ion & #nlev correspond to the initial state and #ion+ & #lev+ correspond to the state to which that ionizes.)

54) Transition probabilities to be computed from quantum defect or as hydrogenic. Transition probabilities most be included as hydrogenic. reals (1; 0.0E+0) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

55) hydrogenic pi xsections, bautista format: i1=lower level; i2=ion;

56) Tabulated Upsilons for HeI from Sawey & Berrington (1993). Every line contains: reals (2n; n log10(temp), n gammas) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

57) Effective ion charge for each level to be used in collisional ionization rates (same as 65) Every line contains: reals (1, Zeff) ; integers (6; N, L, 2*J, Z, #lev, #ion) ; characters (0)

58) Bautista cascade rates (not used)

59) verner pi xc Verner photoionization cross sections, after D. A. Verner & D. G. Yakovlev, 1995, A&AS, 109, 125 r1-r6: fitting parameters ; i1=nuclear z ; i2=number of electrons ; i3=subshell ; i4=verner fitting parameter, orbital quantum number of subshell ; i5=final ion stage-initial ion stage ; i6=final level number ; i7=ion

60) calloway h-like coll. strength Coefficients for analytic fits to Upsilons for H-like ions according to review by Callaway (1994; ADNDT, 57,9) Data lines contain the following information: reals (coefficients) ; integers (4; lower level, upper level, Z, #ion) ; characters (5)

61) Collision strengths from impact parameter approximation (not used)

Every line contains: reals (0); integers (5: dummy, lower level, upper level, Z, #ion); characters (0).

62) calloway h-like coll. strength (same as 60) Coefficients for analytic fits to Upsilons for H-like ions according to review by Callaway (1994; ADNDT, 57,9) Data lines contain the following information: reals (coefficients) ; integers (4; lower level, upper level, Z, #ion) ; characters (5)

63) h-like cij, (hlike ion) Transition probabilities to be computed from quantum defect or as hydrogenic. reals (1; 0.0E+0) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

64) hydrogenic pi xsections, bautista format: i3=z; i1=n; i2=l;

65) effective charge to be used in collisional ionization (h-like ions) Effective ion charge for each level to be used in collisional ionization rates. Every line contains: reals (1, Zeff) ; integers (5; N, L, Z, #lev, #ion) ; characters (0)

66) Kato & Nakazaki fit to collision strengths for He-like ions Like type 69 but in fine structure.

Every line contains: reals (6; fit coefficients) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

67) Effective collision strengths for He-like ions from Keenan, McCann, and Kingston (1987) Every line contains: reals (n; fit coefficients) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

68) Fit to effective collision strengths for He-like ions by Zhang & Sampson. Every line contains: reals (3; fit coefficients) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

69) Kato & Nakazaki (1996) fit to collision strengths for He-like ions. Every line contains: reals (6; fit coefficients) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

70) Coefficients for recomb. and phot x-section of superlevels. Every line contains: reals (#; (den(i),i=1,nd),(Te(i),i=1,nt), (log10(recomb. rates(i,j),i=1,nt,j=1,nd) (ener(i), pi x-secs(i), i=1,nx) ; integers (11; nd, nt, nx, N, L, 2*S+1, Z, #lev+, #nion+, #nlev, #nion) ; characters (0)

71) Radiative transition rates from superlevels to spectroscopic levels The data is for a grid of Ne and Te. Every line contains: reals (#; Ne(i),i=1,nd),(Te(i),i=1,nt), (rad. rates (ne,kt),kt=1,nt,ne=1,nd), Wavelength (Å)) ; integers (6; nd, nt, lower level, upper level, Z, #ion) ; characters (0)

72) Autoionization rates (in s^{-1}) for satellite levels. Every line contains: reals (3; auto. rate, energy in eV above the ionization limit, statistical weight) ;

integers (6; (2S+1), L, #level, continuum level numb., z, ion) ; characters (10; level configuration)

73) Fit to effective collision strengths from Sampson et al. for satellite levels of He-like ions. Every line contains: reals (7; fit coefficients) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

74) Delta functions to add to phot. x-sections to match ADF DR recomb. rates. Every line contains: ; reals (2n+1; ionization limit (eV), (energy over g.s.(i),i=1,n), (amplitude in cm^2(i),i=1,n) ; integers (8; N, L, (2S+1), Z, #lev+, #nion+, #nlev, #nion) ; characters (0)

75) Autoionization data for Fe XXiV satellites : Every line contains: reals (same as 72) (3; auto. rate, energy in eV above the ionization limit, statistical weight) ; integers: lower level, upper ion, upper level, ion ; characters (0)

76) 2 photon decay : Just like data type 50. Every line contains: reals (3; Wavelength (A), gf-val., A-val. (cm-1)) ; integers (4; lower level, upper level, Z, #ion) ; characters (0)

77) Collision transition rates from superlevels to spectroscopic levels Every line contains: reals (#: (Ne(i),i=1,nd),(Te(i),i=1,nt), coll.rates(ne,kt),ne=1,nt,kt=1,nd), Wavelength (Å)); integers (6: nd, nt, lower level, upper level, Z, #ion); characters (0).

78) Level data used for Auger and inner shell fluorescence calculation: Same as type 6:

Every line contains: reals r1=E_Z=1(eV); r2=(2J+1); r3=$n_{effective}$; r4=Ion. Potential; i1=n; i2=(2S+1); i3=L; i4=Z; i5=#lev; i6=#ion) ; c1-c#=configuration.

Different data type used in order to merge with non-Auger levels when assembling database. These data come from the compilation of Kaastra and Mewe (1993).

They are gradually being replaced by more accurate level-to-level data. As of v.21l and later, this has been done for iron and for oxygen. In version 2.1kn4 and earlier, this data was used for all elements.

79) Line data used for Auger and inner shell fluorescence calculation: Same as type 4, but different data type used in order to merge with non-Auger levels when assembling database.

r1=line wavelength (A) ; r2=f value ; i1=lower level index ; i2=upper level index

80) Collisional ionization rates gnd of Fe and Ni : not used

81) Bhatia Fe XIX collision strengths : Every line contains: reals r1=Υ; ; i1=lower level; i2=upper level; i3=#ion);

Energy separation is obtained from level data.

82) Fe UTA rad rates : (from Gu et al. 2006)

Reals: r1=wavelength (A); r2=; r3=gf; r4=$A_{ij}^{radiative}$; r5=A_{ij}^{Auger}; r6=A_{ij}^{total}; i1=lower level; i2=upper level;

83) Fe UTA level data : (from Gu et al. 2006)

Same as type 6:

Every line contains: reals r1=E_Z=1(eV); r2=(2J+1); r3=$n_{effective}$; r4=Ion. Potential; i1=n; i2=(2S+1); i3=L; i4=Z; i5=#lev; i6=#ion) ; c1-c#=configuration.

84) Iron K Pi xsections, spectator Auger binned : No longer used

85) Iron K Pi xsections, spectator Auger summed : Calculates photoionization cross section due to summation of resonances near inner shell edges ala Palmeri et al., 2002 Ap.J.Lett.577, 119.

Every line contains: reals r2=$E_{Threshold}$(Ry); r3=f parameter; r4=γ; r5=scaling factor; i1=lower level; i2=ion;

86) Iron K Auger data : (from Palmer et al. 2003A&A...410..359P, Mendoza et al 2004A&A...414..377M; Palmeri et al., 2003A&A...403.1175P; Garcia et al., 2005ApJS..158...68G)

reals r1=A_{ij} (s^{-1}); i1=#lev; i3=#ion)

88) Iron inner shell resonance excitation : Photoexcitation to autoionizing levels

Format is like types 49: Photon energies are in Ry with respect to the first ionization threshold and cross sections are in Mb. Every line contains: reals (2*np; x(i),y(i),i=1,np) ; integers (8; N, L, 2*J, Z, #lev+, #nion+, #nlev, #nion) ; characters (0) ; (#ion & #nlev correspond to the initial state and #ion+ & #lev+ correspond to the state to which that ionizes.)

B. Utility Programs

The program which translates the ascii database file into the binary fits format used by XSTAR is called bintran.f, and is included with the XSTAR source distribution. Compilation of this program is straightforward, although it requires links to the cfitsio libraries. Execution simply requires the redirection of the input.

Chapter 14

Obtaining and Running XSTAR

XSTAR is available through two sources. It is included in the general FTOOLS distribution (version 4.3 and later) and as a standalone package.

A. XSTAR as Part of the FTOOLS Package

Instructions for installation of the heasoft package are available elsewhere

`http://heasarc.gsfc.nasa.gov/lheasoft`

On a system where heasoft is already installed it is necessary to run the same script required for other ftools in order to make sure environment variables are set properly for xstar:

```
setenv HEADAS /path-to-architecture
source $HEADAS/headas-init.csh
```

where 'path-to-architecture' is the full path to the directory containing the compiled executables and libraries for the headas software.

B. XSTAR as a Standalone Package

The standalone version of xstar is available as a gzipped and tarred file on the xstar website

`http://heasarc.gsfc.nasa.gov/docs/software/xstar/xstar.html`

The source code is available along with compiled executables for several machine architectures. Installation requires that the file be downloaded, and the instructions in the README file followed.

C. Subroutine XSTAR

This is a version of XSTAR which retains most of the functionality of the full code, but which provides a framework whereby XSTAR can be called as a subroutine from another program, or whereby XSTAR can be applied to situations which do not employ the standard assumptions concerning, e.g. the gas density distribution, geometry, or time-steady behavior. This consists of a large fortran file containing all of the subroutines employed by the standard XSTAR, together with two wrapper programs: xstarsetup.f and xstarcalc.f. As implied by the names, xstarsetup.f is intended to be called once at the beginning of a calculation and handles reading of input data and initialization; xstarcalc.f calculates the physical conditions at one point in a photoionized gas and returns level populations, emissivities, opacities, etc. In addition there is a calling program which is intended to illustrate the use of these subroutines. Subroutine XSTAR also requires the two fits data files, atdat.fits and aptrs.fits, and must be linked to the cfitsio subroutine library. All are available via ftp from

`ftp://legacy.gsfc.nasa.gov/software/plasma_codes/xstar/subroutine/`

D. The XSTAR Web Site

See the XSTAR web site http://heasarc.gsfc.nasa.gov/docs/software/xstar/xstar.html for updates and other news about XSTAR.

Chapter 15

Sample Results

A. From "Photoionization and High Density Plasmas", T. Kallman and M. Bautista, June 2000, submitted to Ap. J.

B. Sample Results: Low Density

Although many of the results of XSTAR v.2 calculations are similar to those described in KM82, we present as background some results of simple models which illustrate the behavior of photoionized gas and which disply some of the adopted atomic rates and cross sections. All of the results presented in this and the next section are for optically thin models; we defer a discussion of radiation transfer effects to a later paper.

B.1. Atomic Rates and Cross Sections

We begin by displaying some of the atomic rates which are notable due to their departure from previous work, or to their effects on the model results. Figure 1 shows the ground state photoionization cross sections we adopt. Each panel contains the cross sections for a given element, with various curves for the respective ions. In most cases the various subshells of a given ion are also plotted as separate non-overlapping curves. Resonance structure near threshold of outer shells is apparent, particularly in ions with Z$\geq$10. The photoionization cross sections from many excited levels also show resonance structure. This is illustrated in Figure 2, which shows a few of the excited level cross sections for O VII. Notable are the resonance features near 650 eV, corresponding to the 1-2 transitions in the O VIII ion. Although cross sections with comparable resolution are available for many ions from the opacity project, we adopt Gaussian average fits to these for the great majority of excited levels. For O VII we include all available cross sections at high resolution for ground and excited levels with principle quantum number $n \leq 4$ in order to illustrate the potential importance of the resonance structures in observed spectra.

Ground state collisional ionization rate coefficients are shown in Figure 3. Each panel contains the rates for a given element as a function of temperature.

Figure 4 shows the radiative recombination rates we adopt. We emphasize that these are calculated by performing a Milne integral (equation 3) over the photoionization cross section for each of the bound levels of the recombined ion, and then summing over those rates. This is in contrast to the more typical nebular treatment in which such a sum is fit to an analytic formula as was done by, e.g. Aldrovandi and Pequignot 1973, and has the advantage that it causes all rates to go to detailed balance ratios in the proper limit. Each panel in Figure 4 contains the rates for a given element. Also shown, as the dashed curves, are the rates adopted in XSTAR v.1, i.e. those of Gould and Thakur 1970 (hydrogenic ions) Arnaud and Raymond 1992 (for iron), and Aldrovandi and Pequignot 1973 (all others). Differences are prominent for elements such as C, O, and Fe, and primarily reflect differences between the previous dielectronic recombination rates and those adopted here (e.g. Nahar 2000 and references therein).

B.2. Ionization Balance

In general, the state of the gas depends on both the temperature, via the recombination rates and collisional ionization rates, and on the radiation field, via the photoionization rates. This combined dependence makes a display of the ionization balance cumbersome in the absence of some other simplifying assumption. Figure 5 shows the ionization balance in the coronal case, i.e. when the radiation field is negligible, as a function of temperature. This can be compared with other similar calculations such as, e.g. Arnaud and Rothenflug 1985.

B.3. Heating and cooling rates

A by-product of the ionization and excitation balance is the emissivity and opacity of the gas, which correspond to the net heating and cooling rates. Figure 6 shows the heating and cooling rates as a function of temperature and ionization parameter for the various elements. Heating rates are shown as solid curves, cooling rates as dashed curves. Rates assume solar abundances (Grevesse, Noels and Sauval 1996), and are given in units of erg s^{-1} cm^{+3} per H nucleus. Different curves correspond to ionization parameters log(ξ)=0,1,2,3,4 for an ε^{-1} power law ionizing spectrum Fewer curves appear in some panels owing to pile up at low ionization parameters for elements such as carbon, while for H and He, the log$\xi \geq 2$ curves fall below the range plotted. These are calculated in the limit of low gas density, n=1 cm^{-3}.

A coronal plasma cools more efficiently, in general, than a photoionized plamsa since the ionization state is lower at a given temperature. Figure 7 shows the cooling rate as a function of temperature for such a plasma. Comparison of these rates with the results of Figure 6 shows similarity with the cooling rate at the lowest ionization parameter plotted there (logξ=0), although the coronal rates are generally larger at low temperatures. This is a reflection of the fact that at logξ=0 there is significant photoionization of the neutral and near-neutral species.

B.4. Thermal Balance Calculation

When the condition of thermal equlibrium is imposed, then the temperature is determined as a function of ionization parameter for a photoionized plasma. Figure 8 shows the ionization and temperature of an optically thin low density photoionized gas with a ε^{-1} ionizing continuum, as a function of ionization parameter. This can be compared with the results of KM82, model 7 (although that model was not optically thin for $\log(\xi)\leq 2$). The current model is significantly more highly ionized; the ionization parameter where the abundances of O VII and VIII reach their maximum is lower by 0.5 dex in the current models. The temperature calculated here is lower; this may be due to a different choice of low and high energy spectral cutoffs which affect the Compton equilibrium temperature.

C. Sample Results: High Density

At high densities, various physical processes become important which can affect the ionization and thermal balance. These include:

Lowering of the continuum, in which collisional ionization from highly excited levels (i.e. superlevels) results in a net reduction in the effective reccombination rates. This effect is only included for H and He-like ions in our calculations. This process is most important at low temperatures. A competing effect is collisional deexcitation from superlevels, but this turns out to be less important than continuum lowering.

3-body recombination results in a net increase in total recombination rate. In our models we include collisional ionization and 3-body recombination from essentially all levels, although this process is generally more important for levels closest to the continuum.

At high densities the spectroscopic level populations in the recombining ion can approach their LTE values, leading to enhanced collisional ionization from these levels and a decrease in the total recombination. This turns out to be unimportant for most ions at the densities and temperatures we consider.

In a photoionized plasma the incident photon flux must be very large if the density is high and the ionization parameter is within the conventional range. Such high photon fluxes can lead to large enhancements in the recombination rate via stimulated recombination.

C.1. Density dependent recombination rates

The effects of density on recombination rates are illustrated in figure 9, which shows recombination rate as a function of density for various ions of H, He and O. The curves correspond to temperatures logarithmically spaced between 10^4K and 10^7 K, and the dashed curve shows the XSTAR v.1 value. Stimulated recombination can cause large enhancements in the rates at high density, but its effects are dependent on the shape of the assumed ionizing spectrum; therefore it has been excluded from the results until the end of this section for illustrative purposes. In the H and He-like ions the lowering of the continuum is apparent at moderate densities, and the effect of 3-body recombination is apparent in H I and He I at the highest densities. This is an illustration of the fact that, at a given density, 3 body recombination is greater for ions with lower free charge, and is greater at lower temperature (Bautista and Kallman 2000). The calculations shown in this figure were done at a fixed ionization parameter of log(ξ)=2 with an ε^{-1} ionizing continuum. As a result the lower ionization stages, of oxygen, O I, O II, and O III have low abundance and are not included in the full multilevel matrix calculation and their recombination rates are treated using the total rates shown in figure 3. Other ions show the effects of continuum lowering, which causes a decrease in rate by a factor of up to $\sim$ a few beginning at densities greater than 10^6–10^8.

C.2. Level populations vs. density

In addition to enhancing the total recombination rate, high densities enhance the importance of collisional processes relative to radiative processes in bound-bound transitions. Level populations approach their LTE values, which may greatly exceed the recombination values for levels with dipole allowed decays. This is illustrated in figure 10, which shows the ratio of level populations to LTE populations (departure coefficients) as functions of density for H I at $\log(\xi)$=-5 and T=10^4K (panel a) and for O VIII at $\log(\xi)$=-5 and T=10^6K (panel b). Departure coefficients of all bound spectroscopic levels decrease proportional to density, approaching assymptotic values at densities greater than 10^{17} cm^{-3}. The superlevels exhibit slower dependence on density, reflecting the fact that they are likely to be in LTE with the continuum at lower density than the spectroscopic levels.

C.3. Heating-cooling vs. density

Heating and cooling rates depend on density via the ionization balance and via the rates for the heating and cooling per ion. In the previous subsection we have shown that at the highest densities we consider the recombination rates can be enhanced by 3 body recombination, or reduced, by continuum lowering and collisional ionization. The former process is dominant at high densities for H I and He I, while the latter dominates for moderate densities for other ions. Since the thermal balance in a photoionized gas is dominated by H and He when the ionization parameter is low (i.e. $\log(\xi) \leq 1$), and by more highly charged ions at higher ξ, we expect the heating and cooling to be affected differently at high densities in the two different regimes. Although the dependence of cooling rate on ionization balance at low densities is not generally monotonic (c.f. figure 6), for many ions the heating rate is greater at lower ionization parameter. The per ion heating rate depends on the photon flux rather than the gas density, while the per ion cooling rate is suppressed by collisional deexcitation. Figure 11 shows the dependence of heating and cooling rates on density and temperature, in a form analogous to that of figure 10. Curves show cooling (dashed) and heating (solid) rates at 5 temperatures spaced logarithmically between between 10^4K and 10^7 K, for log(ξ)=2 and a ε^{-1} power law ionizing spectrum. For highly ionized species heating rates are decrease slightly with density, while cooling rates increase. H and He I behave in the opposite way, owing to the increase in recombination (which increases the neutral fraction and hence the photoionization heating) and to the collisional suppression of radiative decays (which decreases the net radiative cooling).

C.4. Thermal Equilibrium

Figure 12 shows the results of a thermal balance calculation of an optically thin photoionized gas as a function of density and ionization parameter. The curves correspond to ionization parameters log(ξ)=4,3,2,1 for the same power law ionizing spectrum used previously. This demonstrates that the net effect of higher densities is an increase in temperature at the highest densitied and lowest temperatures, and a decrease in temperature at lower density and higher temperature, for the reasons listed in the previous section. We emphasize that the quantitative value of the temperature, particularly for high ionization parameters and/or temperatures greater than 10^7 K or so, depends on the detailed shape of the ionizing spectrum over all energies, owing to the possible importance of Compton heating and cooling (the spectral dependence of the effects of stimulated recombination, again, have been excluded from these results).

C.5. Ionization distribution, high n

Figure 13 shows the ionization and thermal balance of an optically thin photoionized gas analogous to that shown in Figure 8, but at a density of 10^{17} cm^{-3}. Comparison shows that the high density results in generall higher ionization ste for most elements at high ionization parameter, owing to the reduction in the net recombination rate. At the same time, the temperature is slightly lower, as described in the previous subsection. The opposite is true at the low ionization parameter extreme of figure 13 – the temperature is slightly greater than in figure 9 due to the enhanced recombination rates of H and He I at high densities.

C.6. Warm absorber, high density

High densities also affect the absorption and emission spectra of photoionized plasmas. Figure 14 shows a comparison of a warm absorber spectrum at densities of 10^4 cm^{-3} (panel a) and 10^{17} cm^{-3} (panel b) due to oxygen in the 0.5-1 keV energy range. The ionization parameter is log(ξ)=2 and the temperature is 10^5K. In the high density case there is a prominent absorption structure near 0.65 keV, associated with resonances in the photoionization cross sections from the 1s2s configurations in OVII. These resonances are apparent in the cross sections shown in figure 2, and they appear in opacity due to the build-up of excited level populations at high densitiies. Such features are potentially observable in the spectra of astrophysical X-ray sources such as the partially ionized absorbers associated with Seyfert galaxies (e.g. George et al. 1998) if these objects contain gas at densities comparable to those considered here.

C.7. Recombination emission, high density

Figure 15 shows a comparison of the emission spectrum at densities of 10^4 cm^{-3} (panel a) and 10^{17} cm^{-3} (panel b) due to oxygen in the 0.5-1 keV energy range. The ionization parameter is $\log(\xi)=2$ and the temperature is 10^5K. In the high density case the ratio of continuum to line emission is reduced, and the ratios of the He-like lines is changed from the familiar low-density case in which the forbidden/intercombination line ratio is large at low density to values $\sim$1 at high density.

C.8. The effects of stimulated recombination

So far in this section we have artificially excluded the effects of stimulated recombination (by manually setting the rates to zero when calculating total recombination). We illustrate the effect of relaxing this condition in figure 16, which is the equivalent of figure 9 (recombination rates vs. density) but with stimulated recombination included. Again the ionizing spectrum is a ε^{-1} power law, which has strong flux at the lowest photon energies. Comparison of figures 16 and 9 shows that the rates are greatly enhanced at high densities, and this enhancement is greatest for ions with lowest ionization potentials. This is due to the influence of the low energy photons on the stimulated recombination rate, and a different spectral shape (e.g. a blackbody) would produce a different distribution of recombination with charge state at high densities.

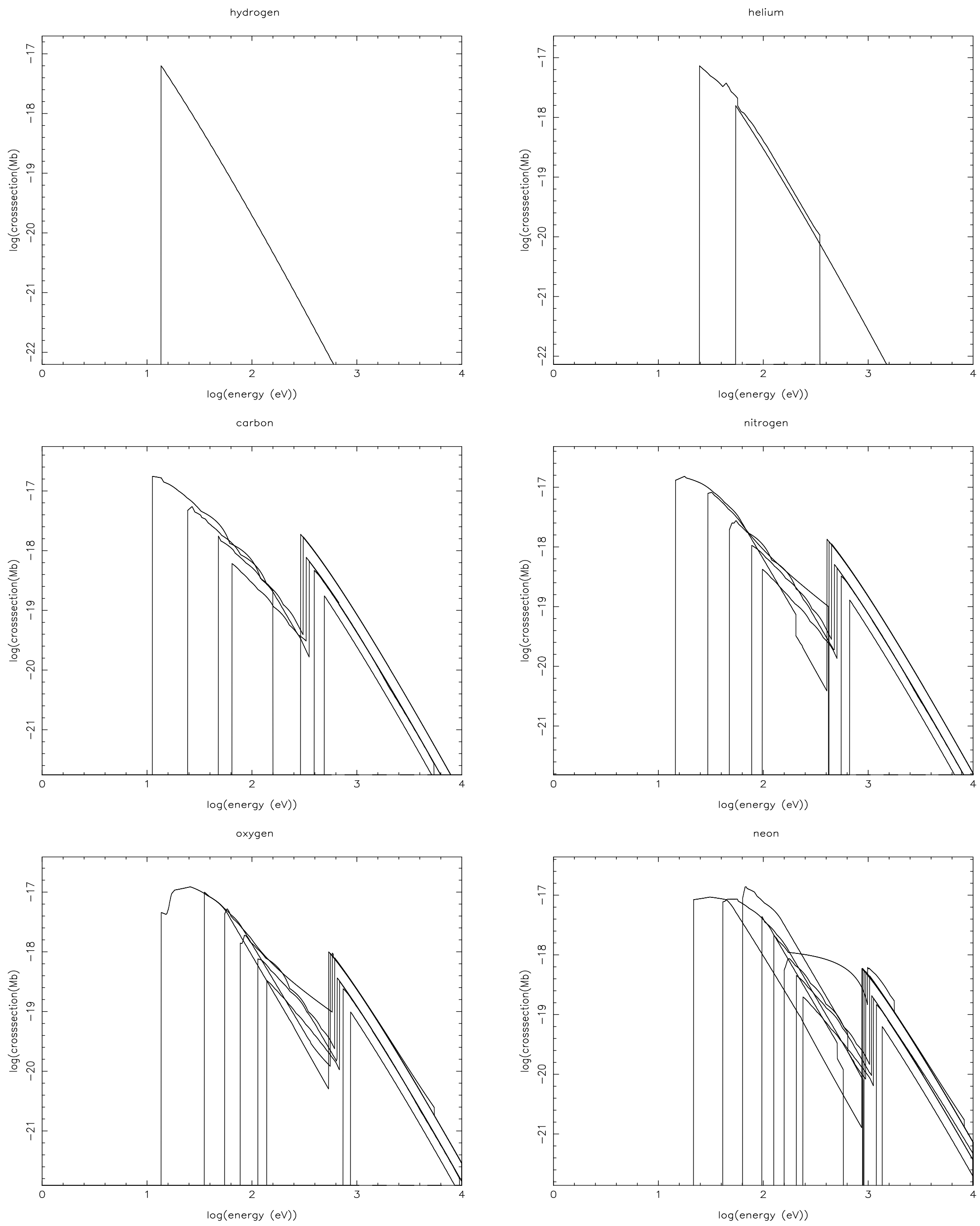

Fig. 1.— figure 1a

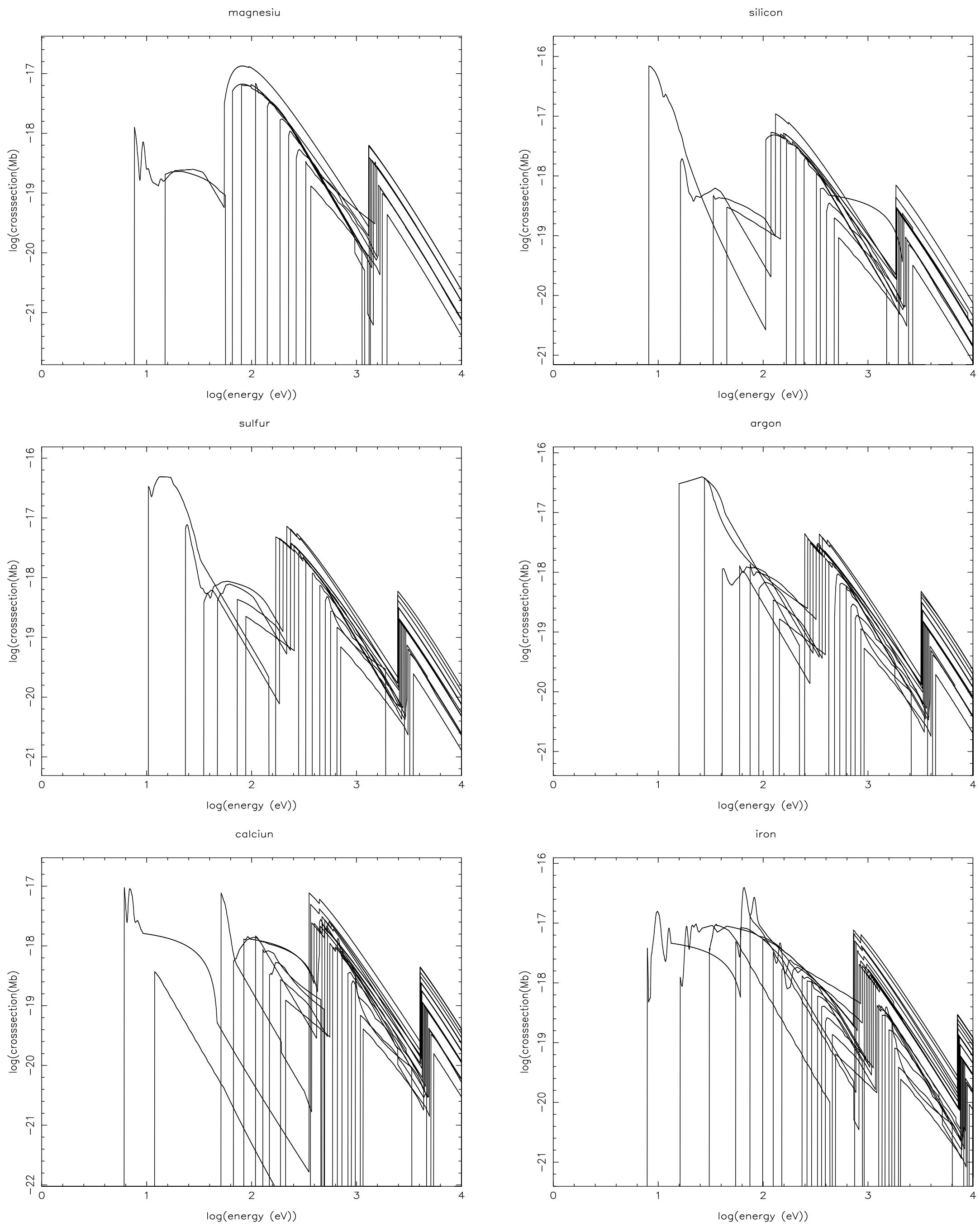

Fig. 1.— figure 1b

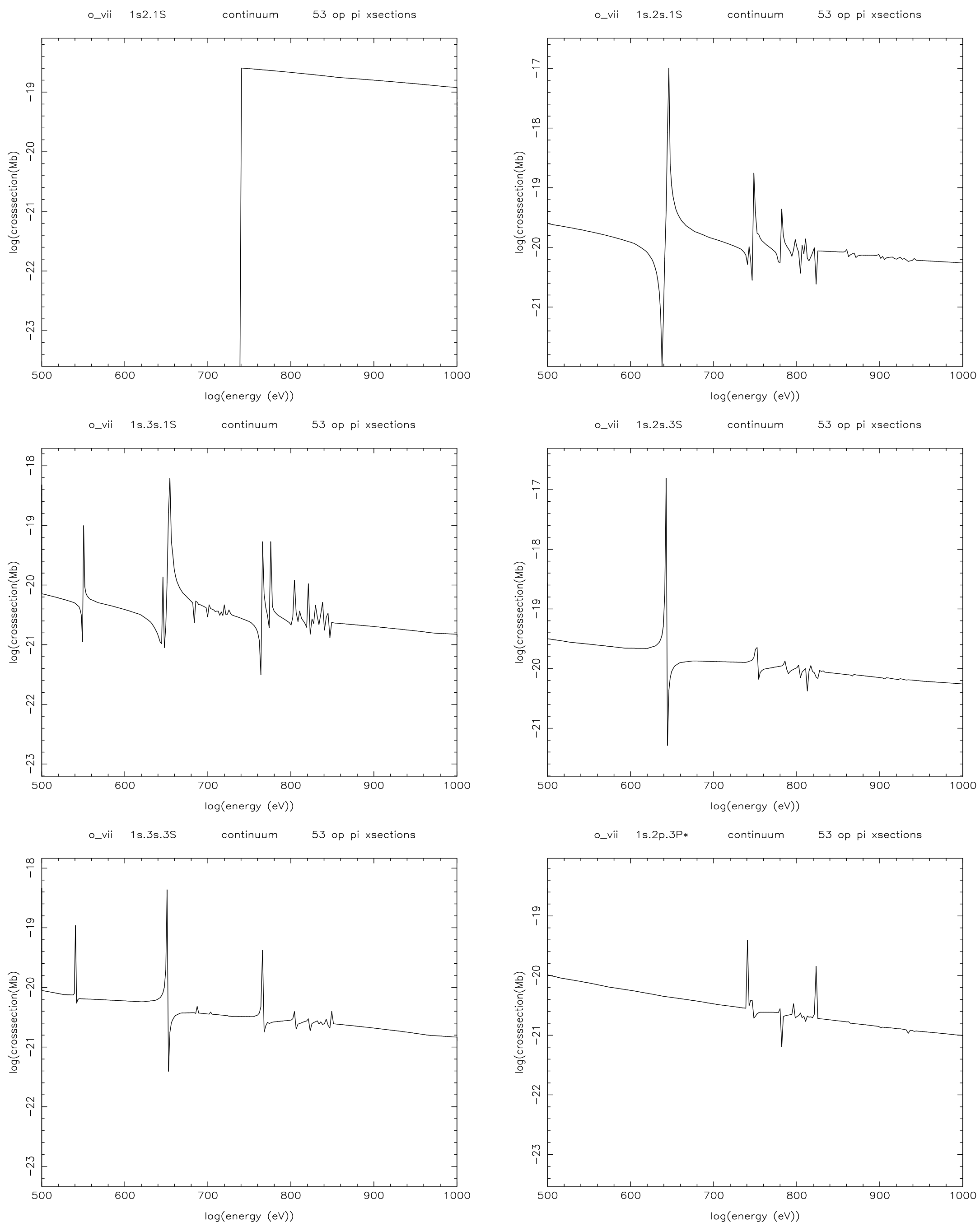

Fig. 2.— figure 2

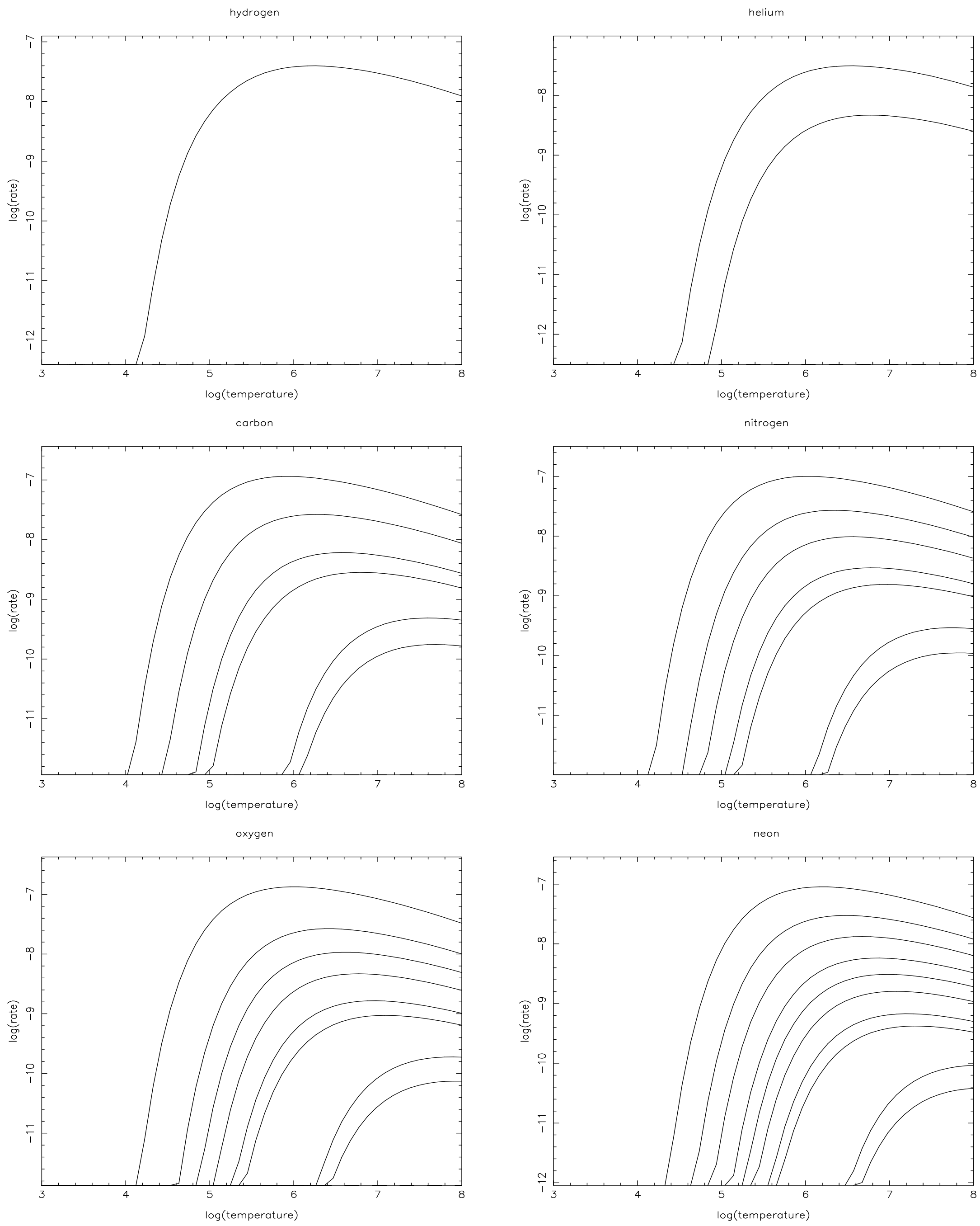

Fig. 3.— figure 3a

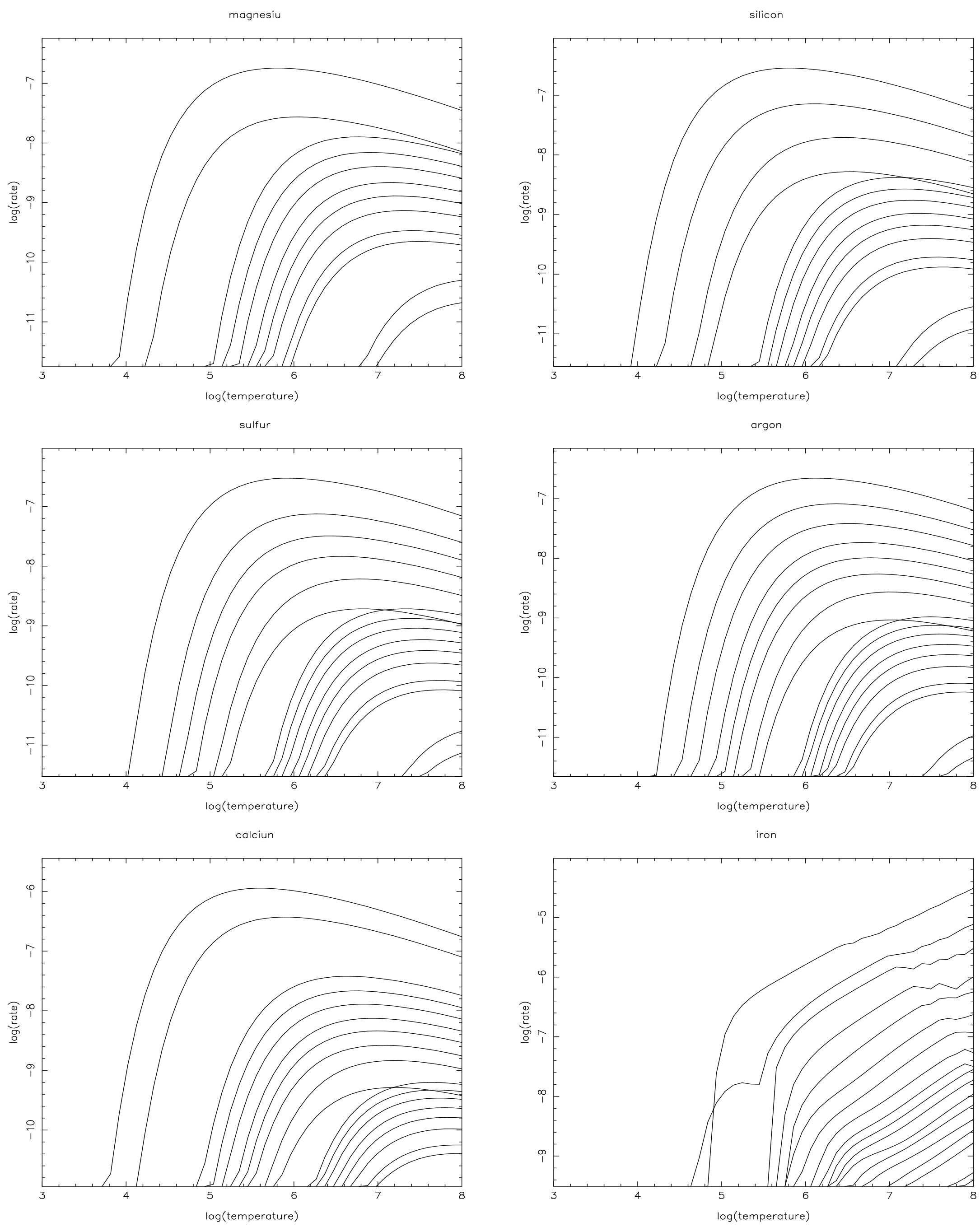

Fig. 3.— figure 3b

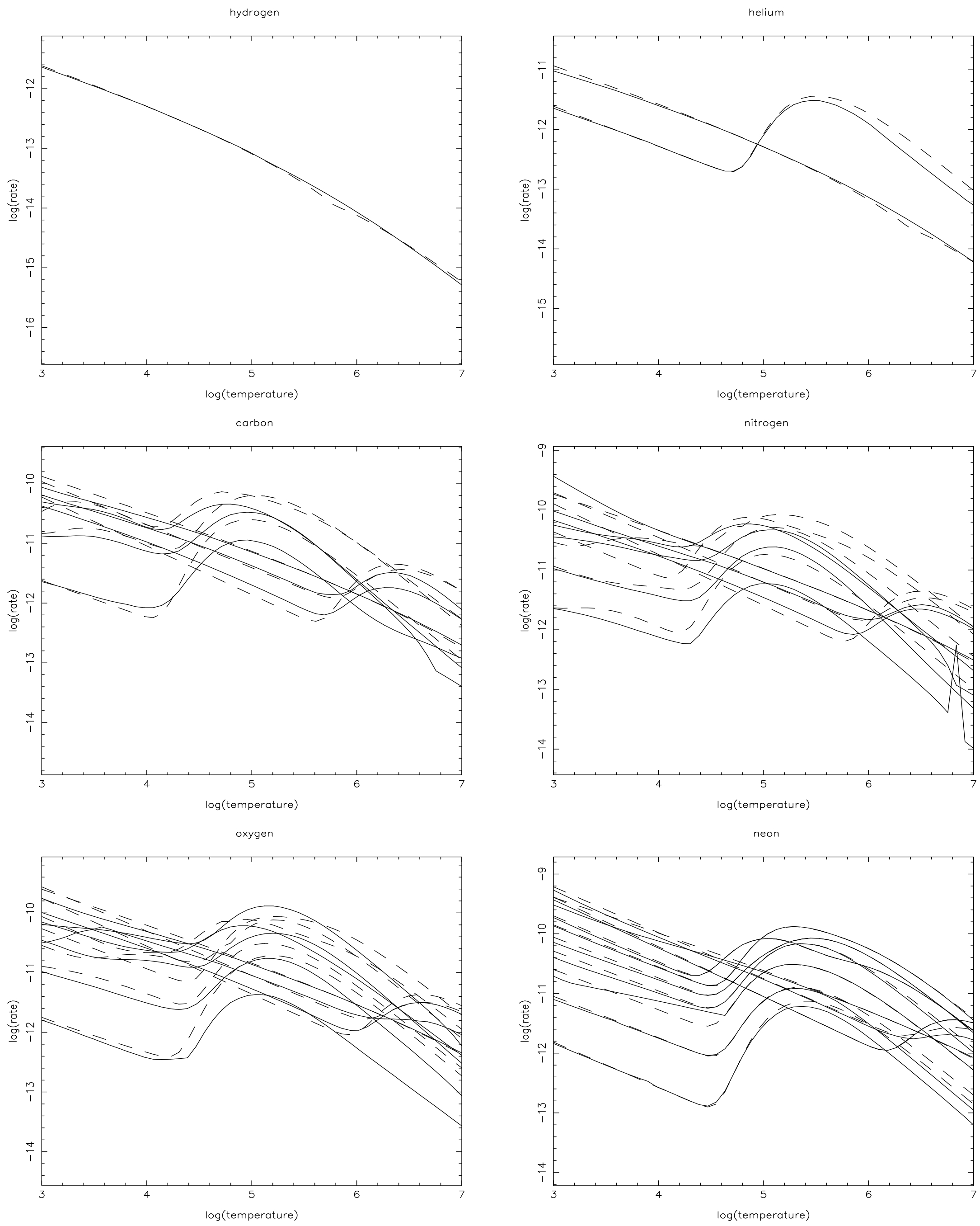

Fig. 4.— figure 4a

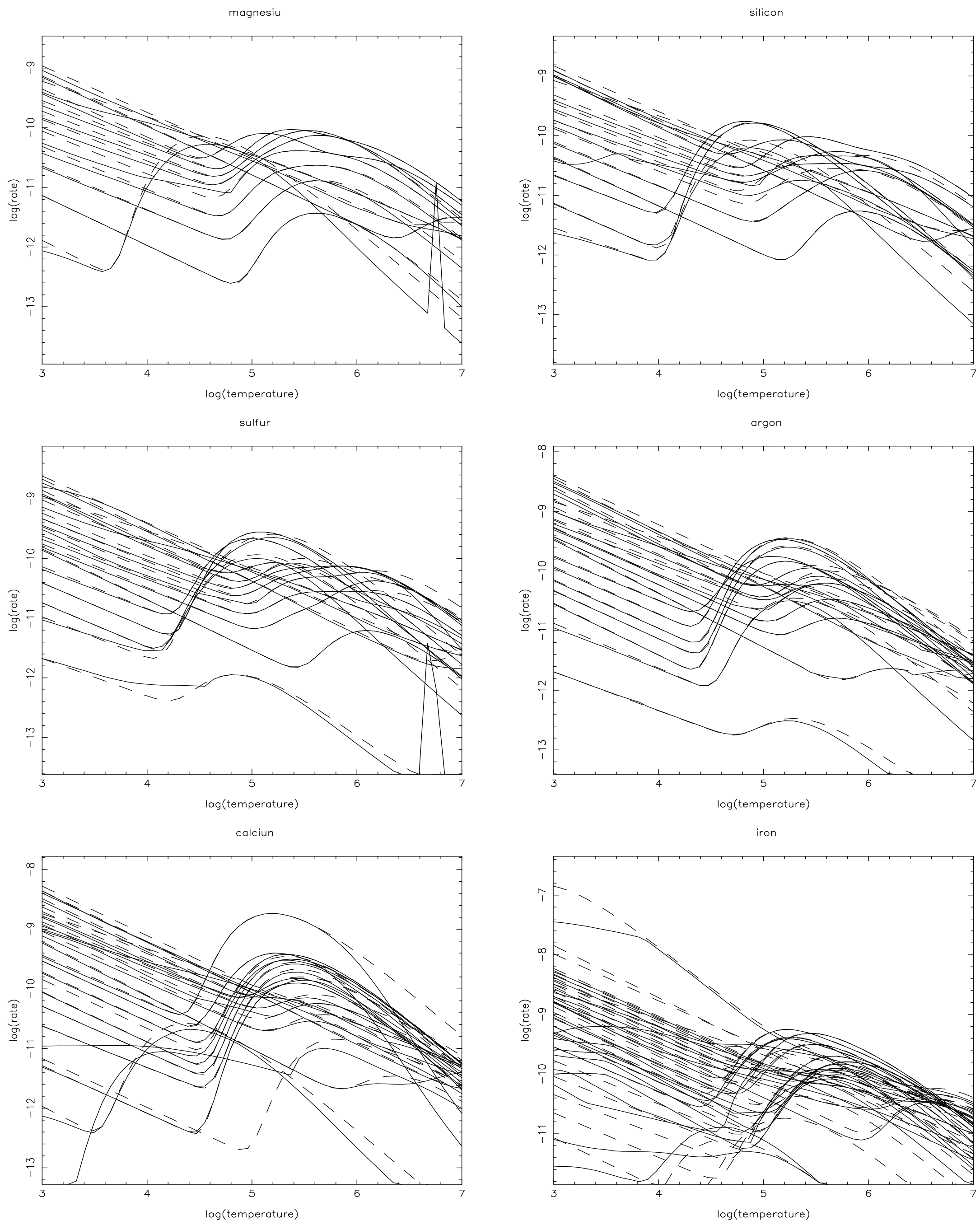

Fig. 4.— figure 4b

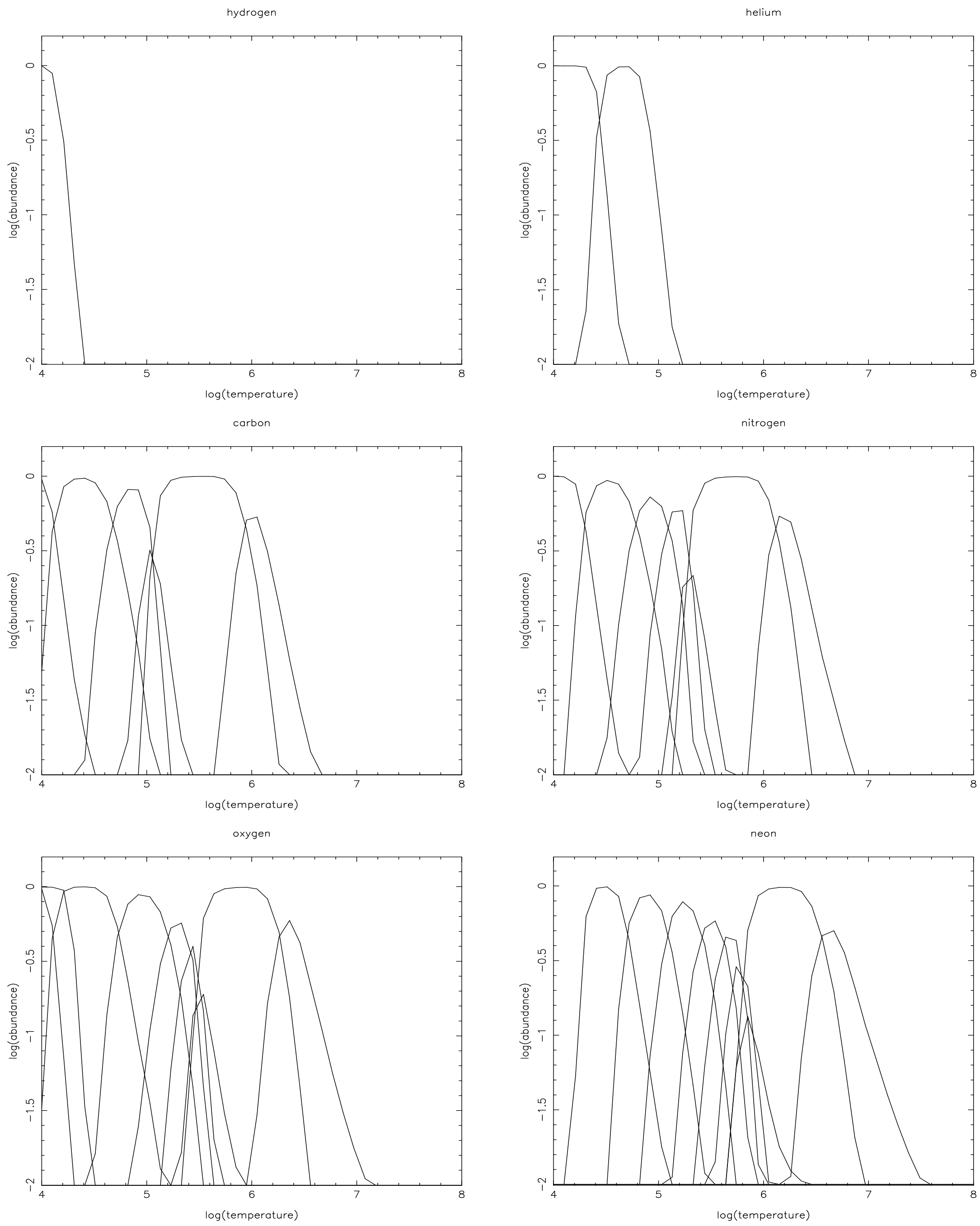

Fig. 5.— figure 5a

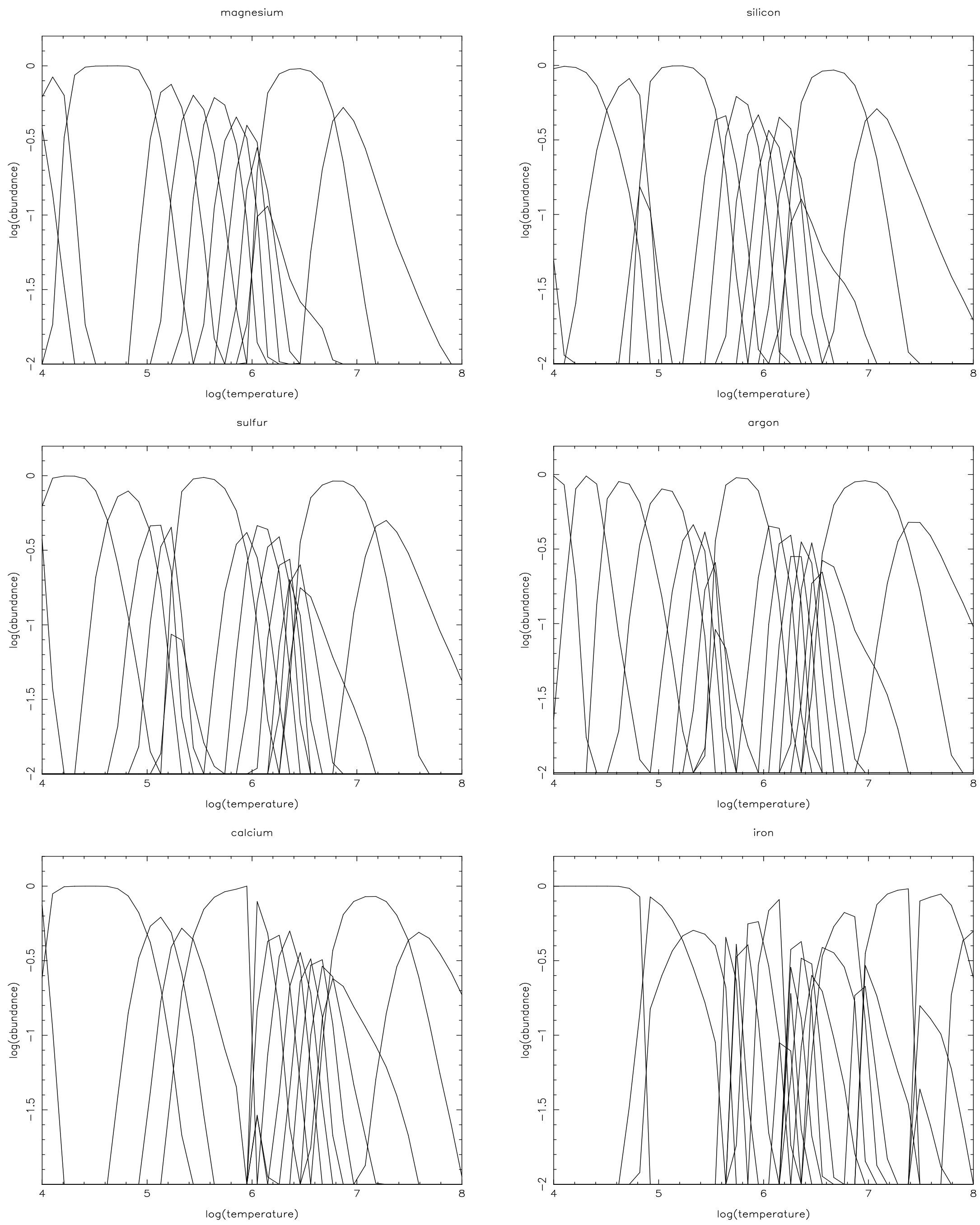

Fig. 5.— figure 5b

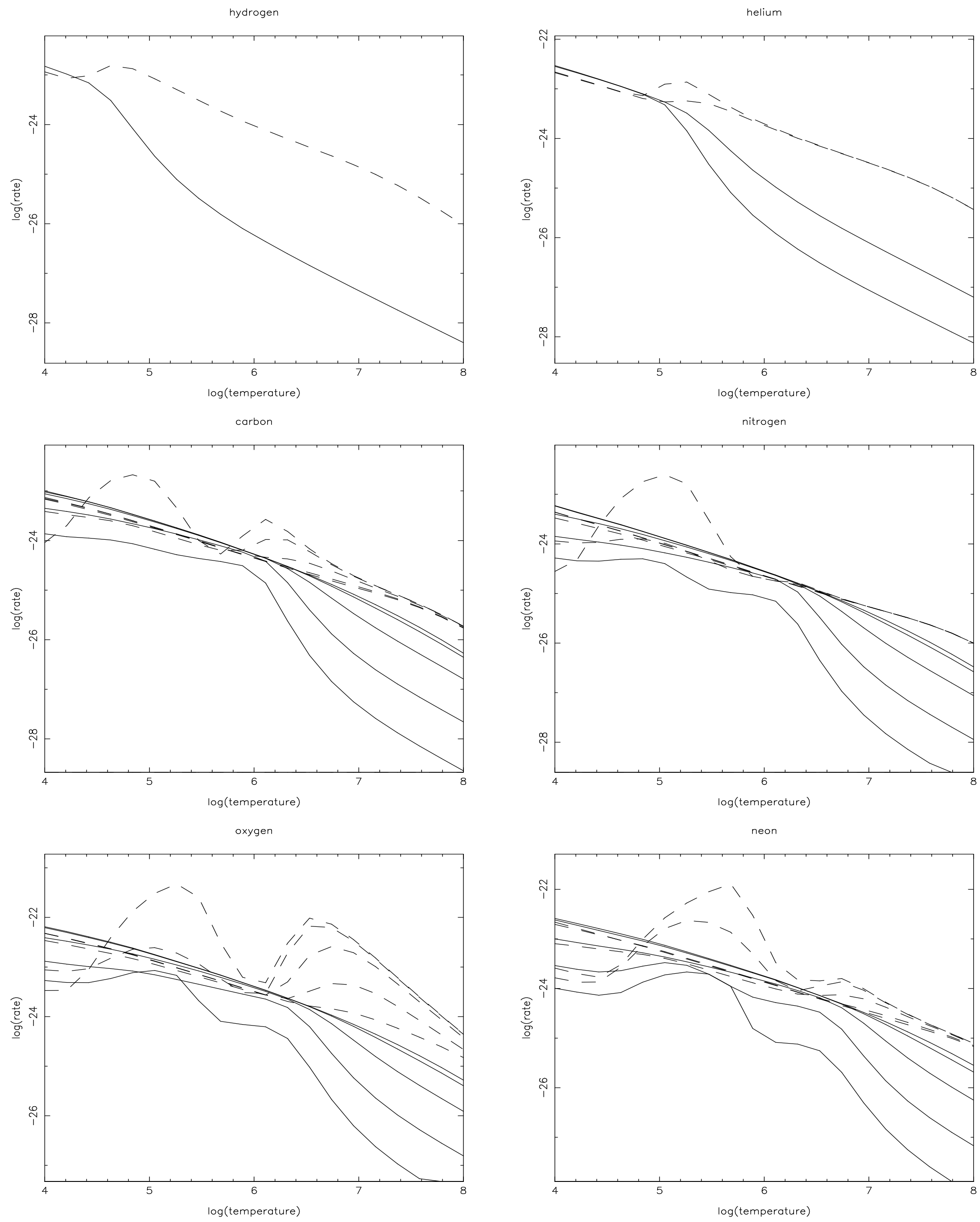

Fig. 6.— figure 6a

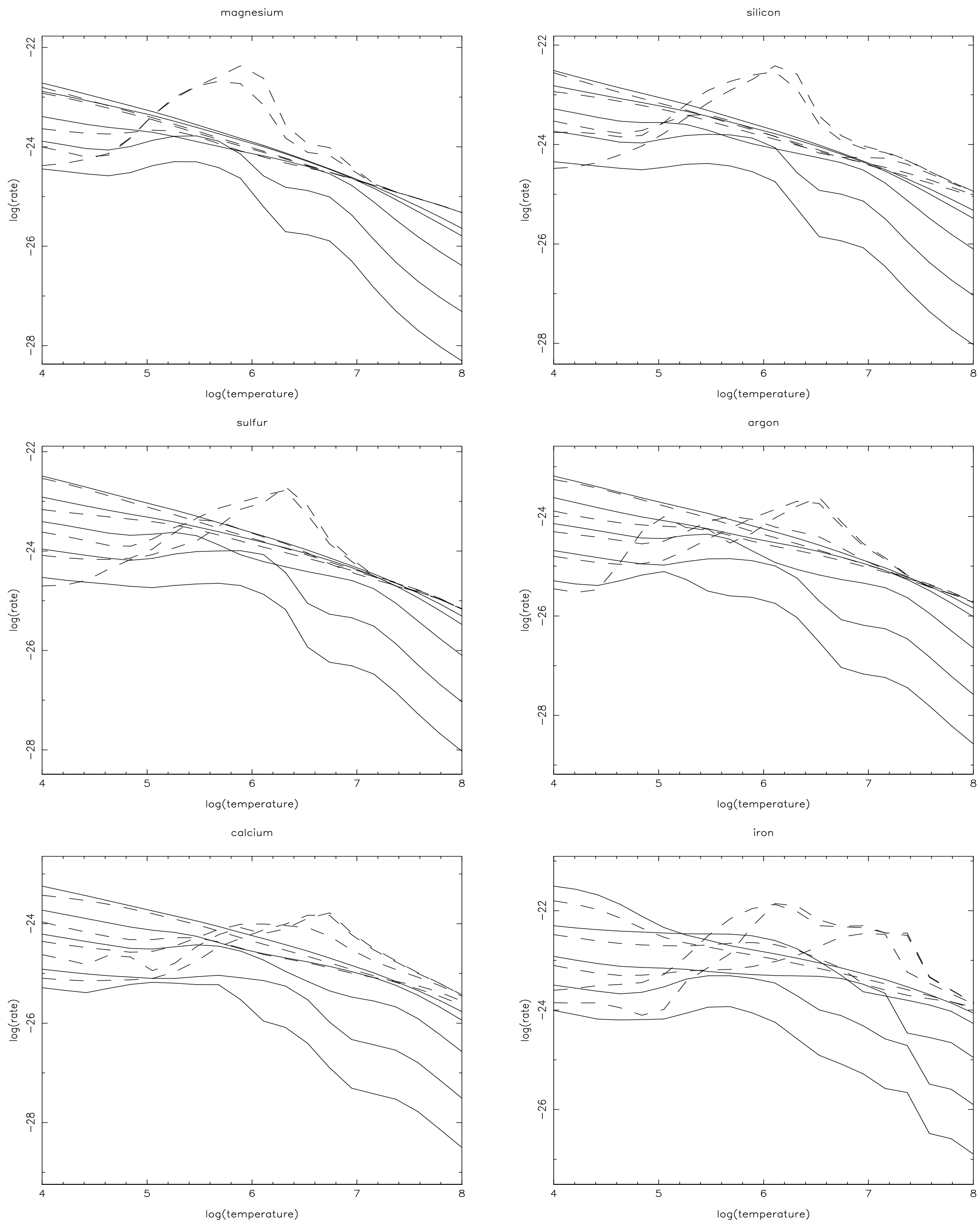

Fig. 6.— figure 6b

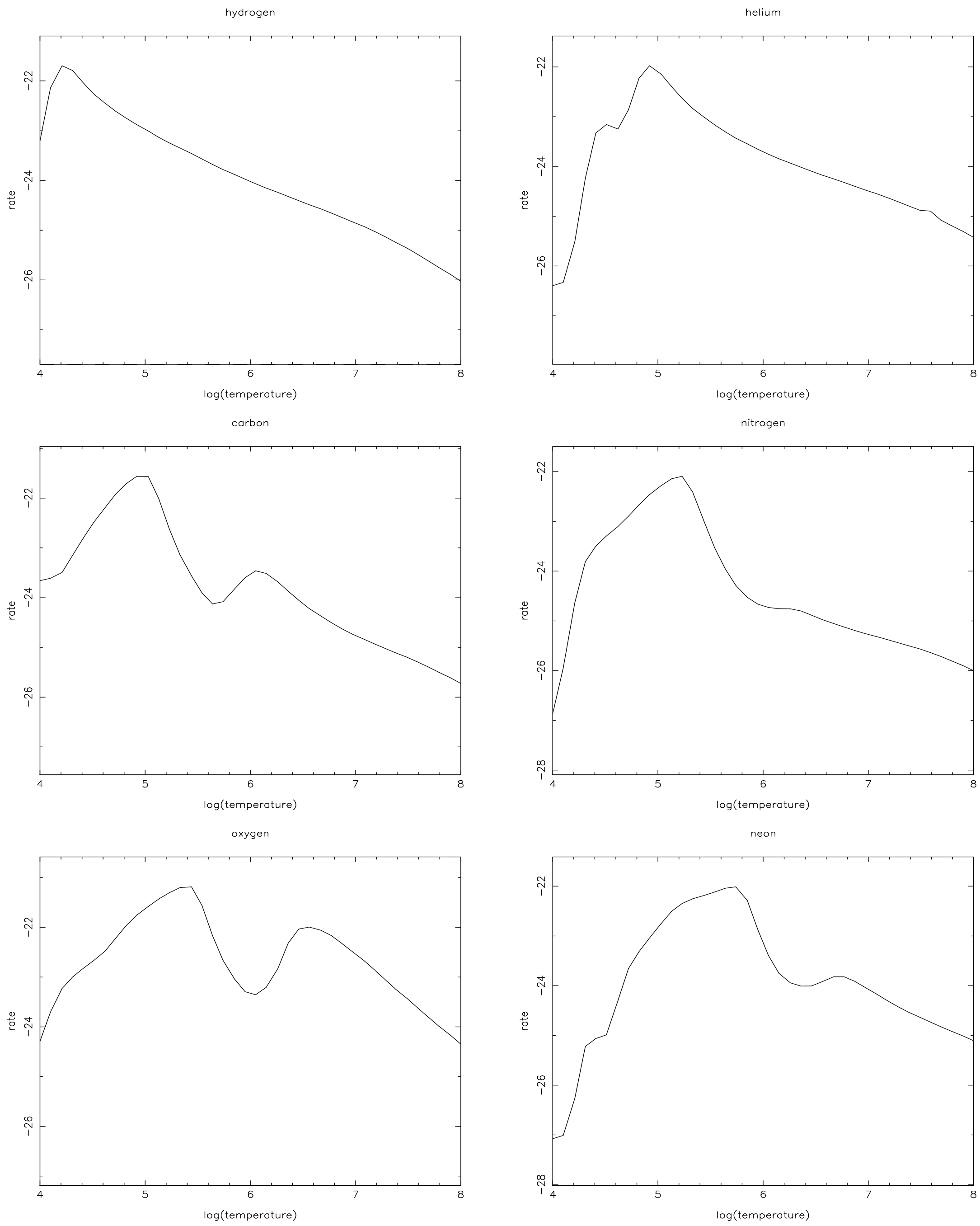

Fig. 7.— figure 7a

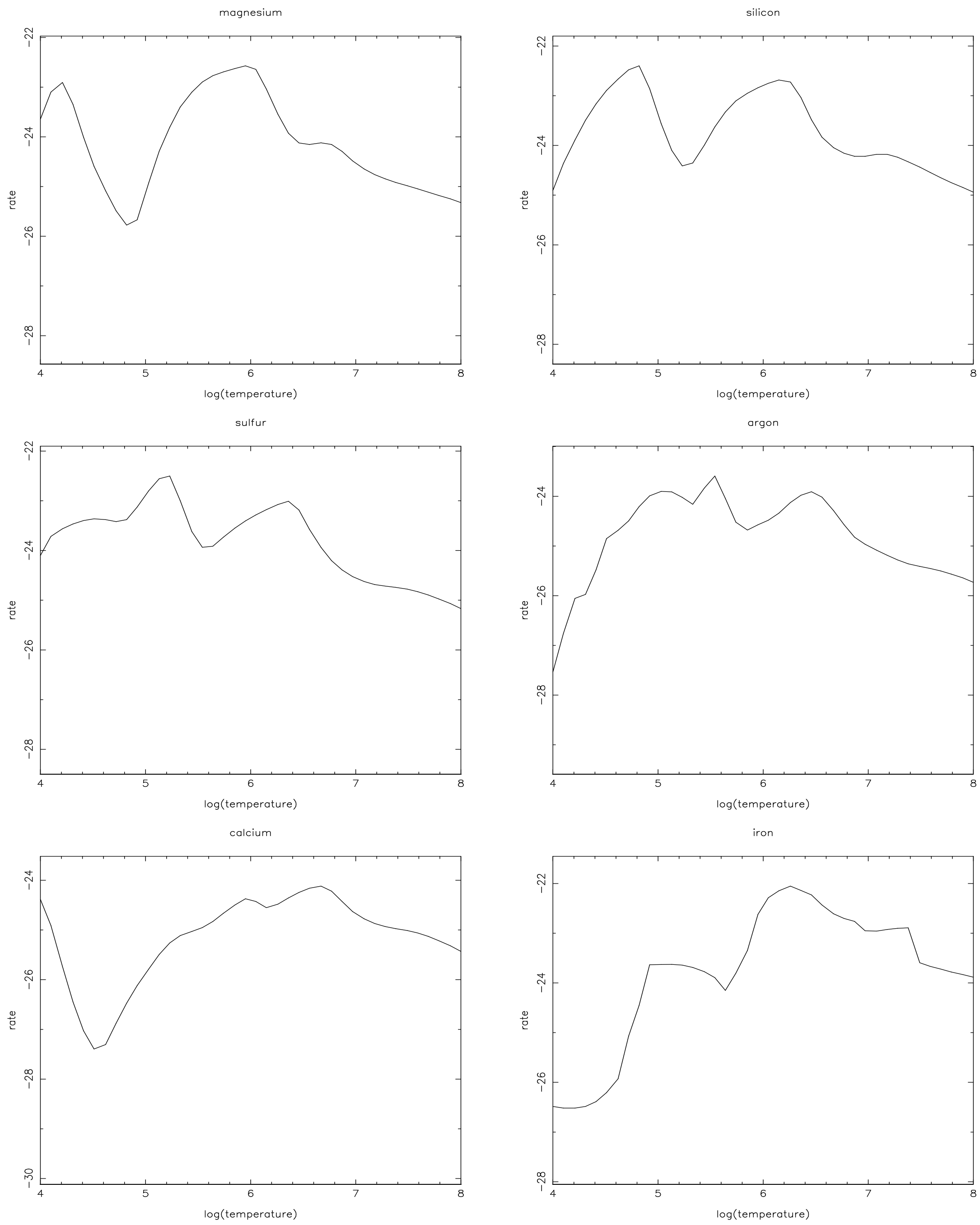

Fig. 7.— figure 7b

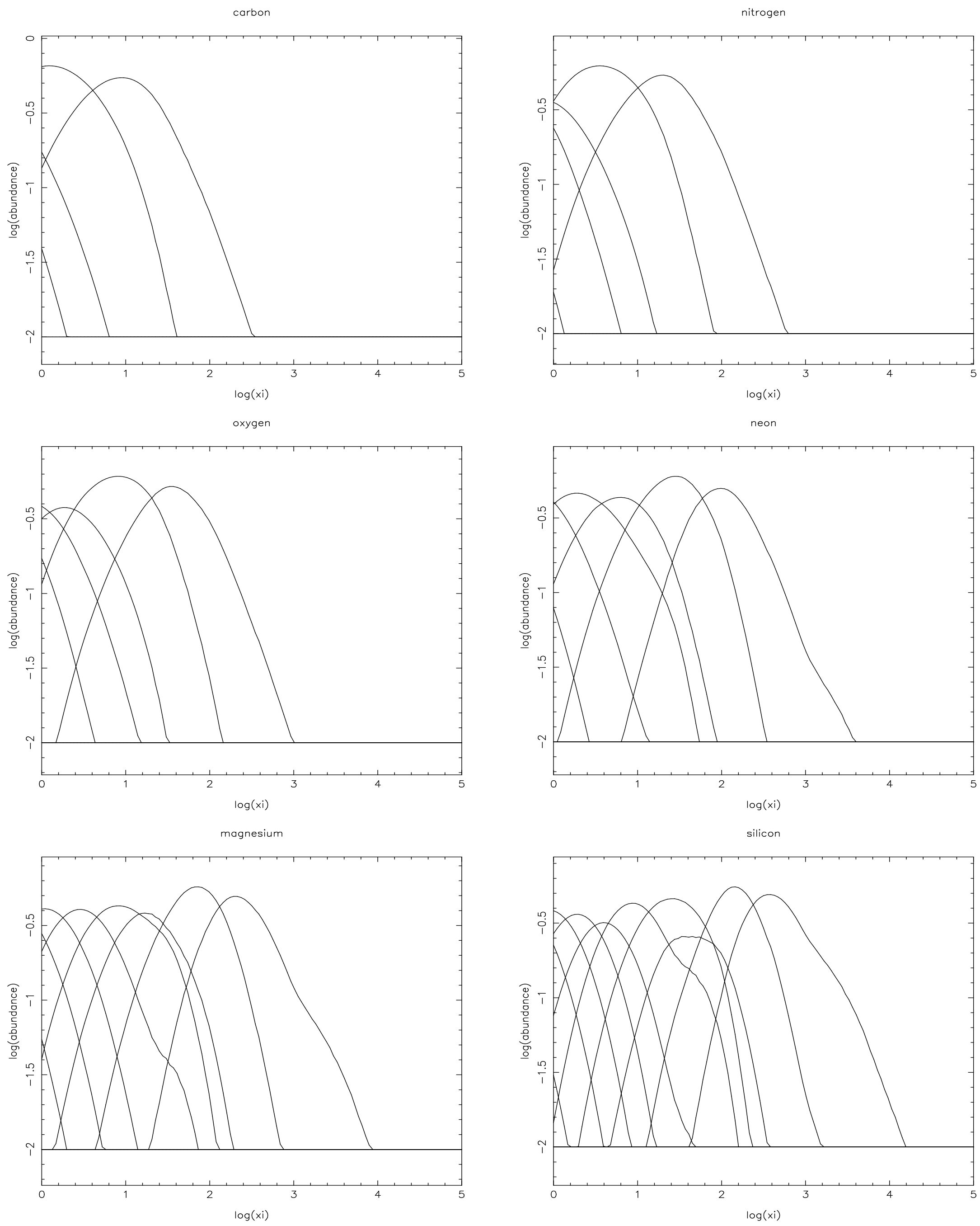

Fig. 8.— figure 8a

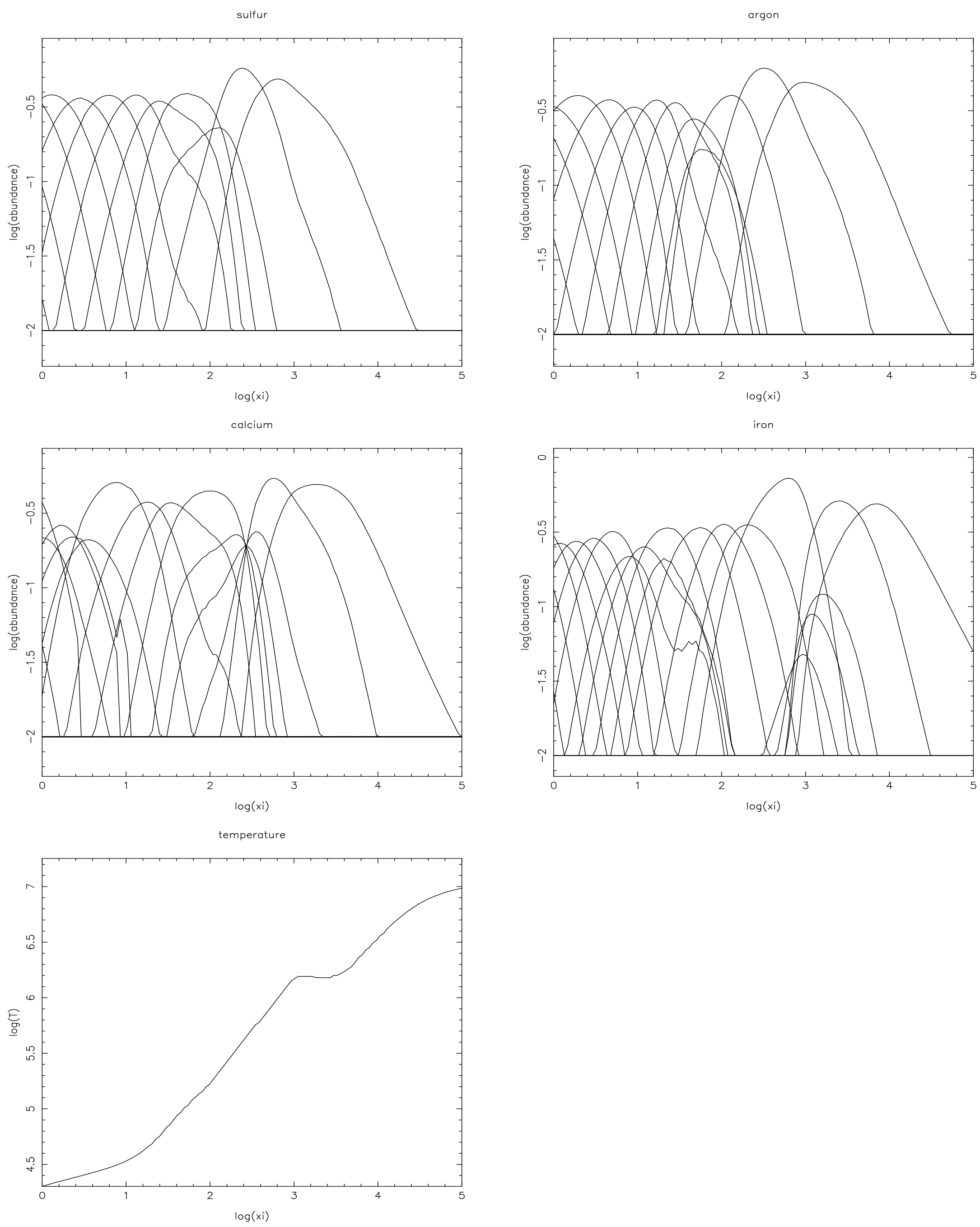

Fig. 8.— figure 8b

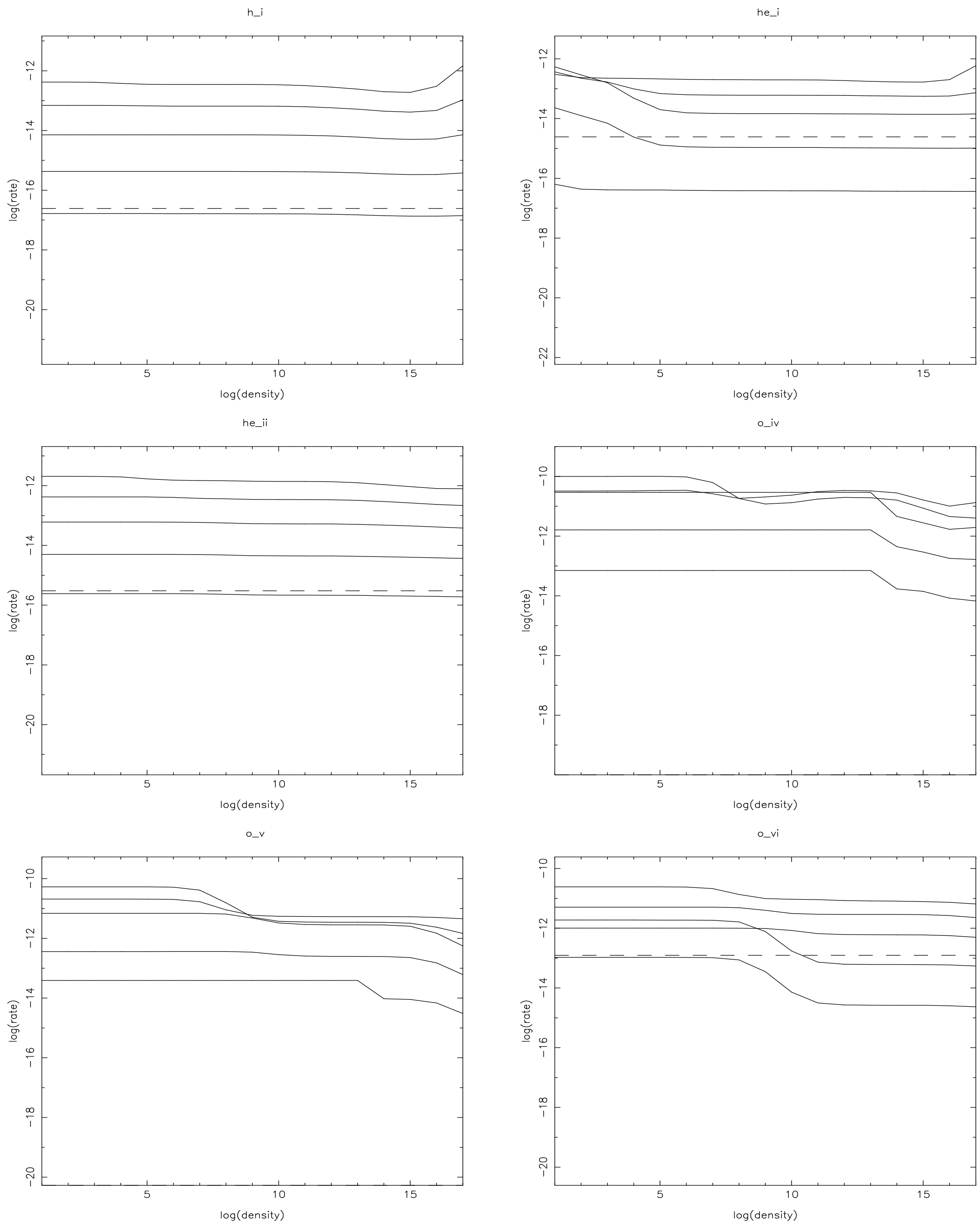

Fig. 9.— figure 9a

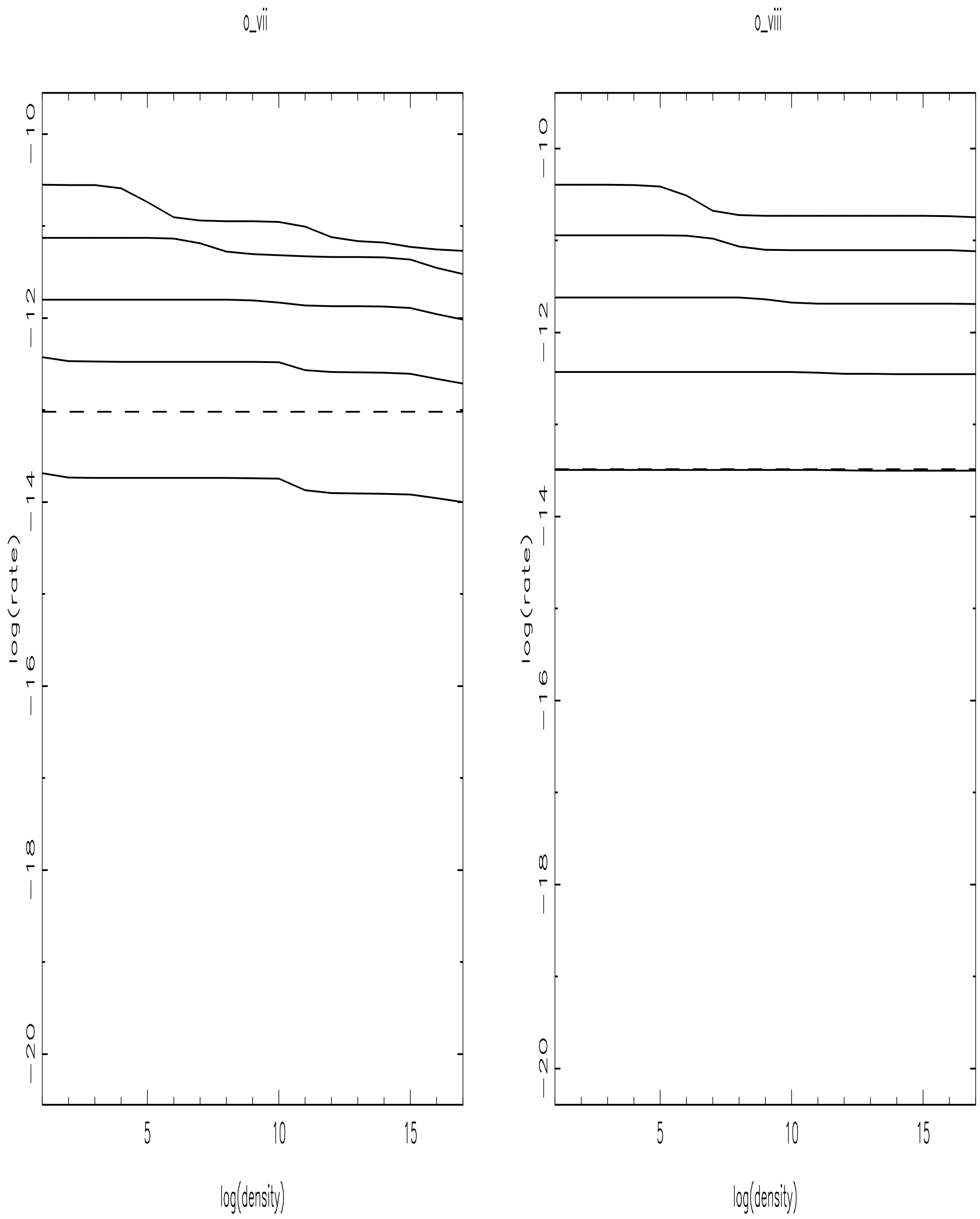

Fig. 9.— figure 9b

Chapter 16

Sample Results, Continued

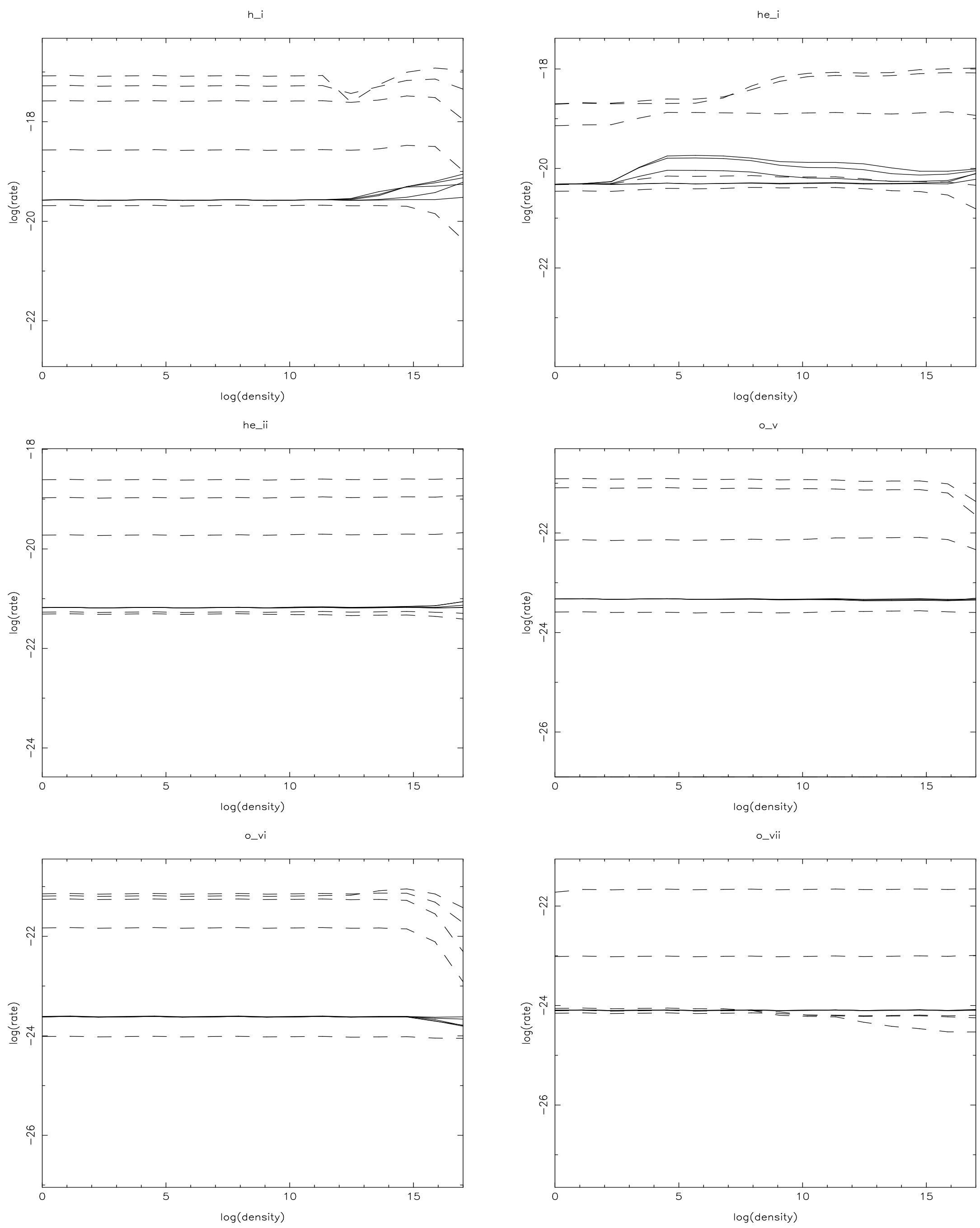

Fig. 11.— figure 11a

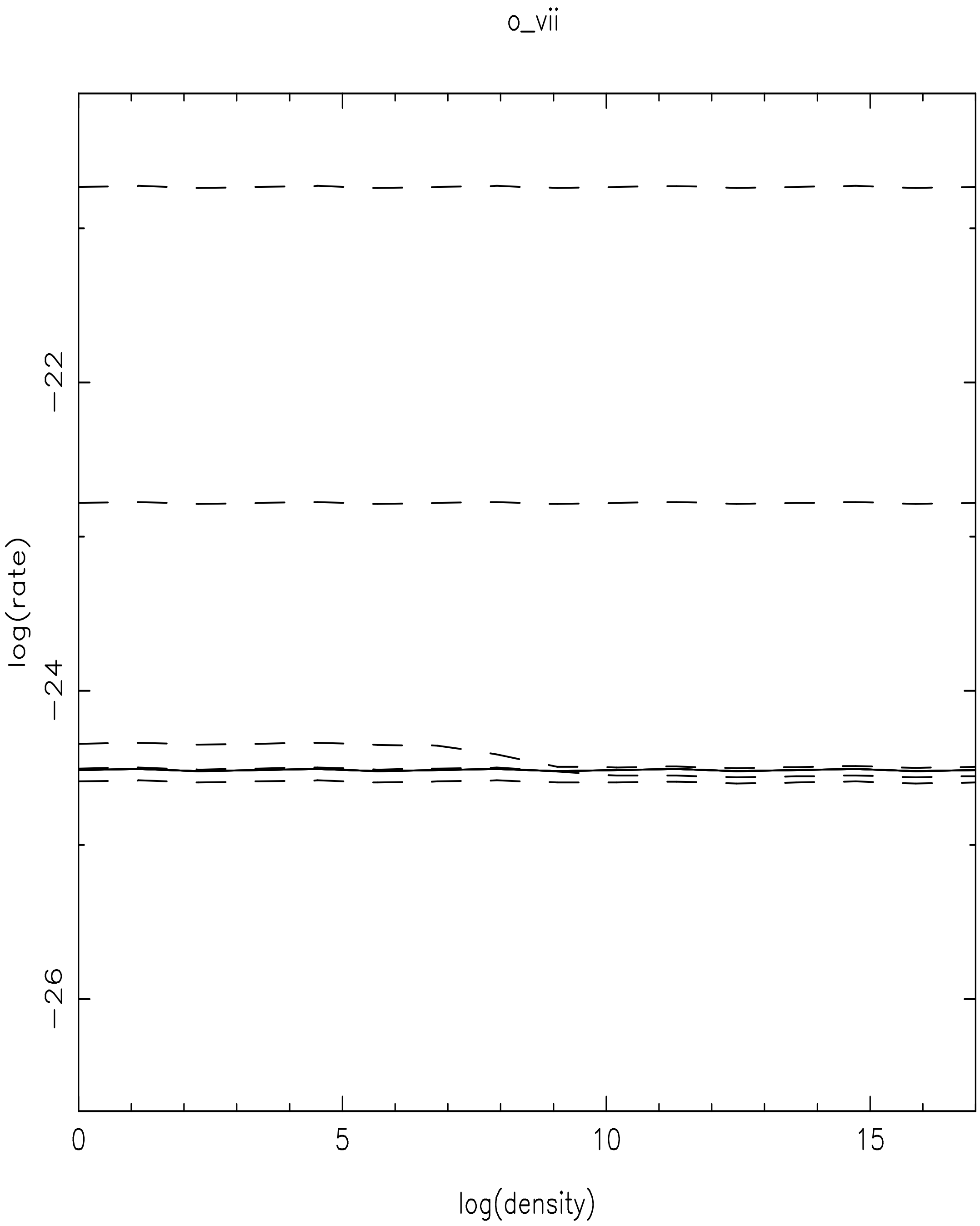

Fig. 11.— figure 11b

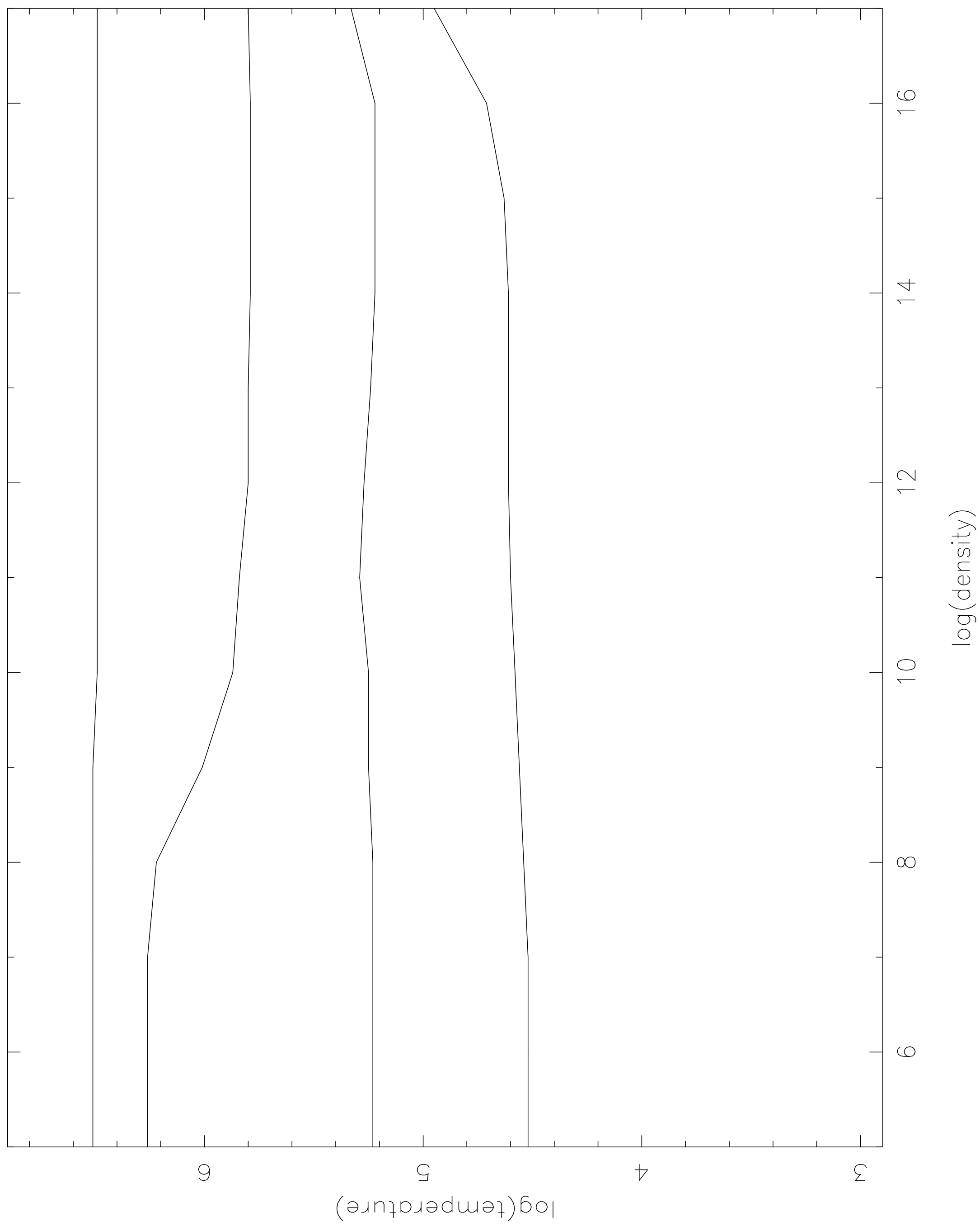

Fig. 12.— figure 12

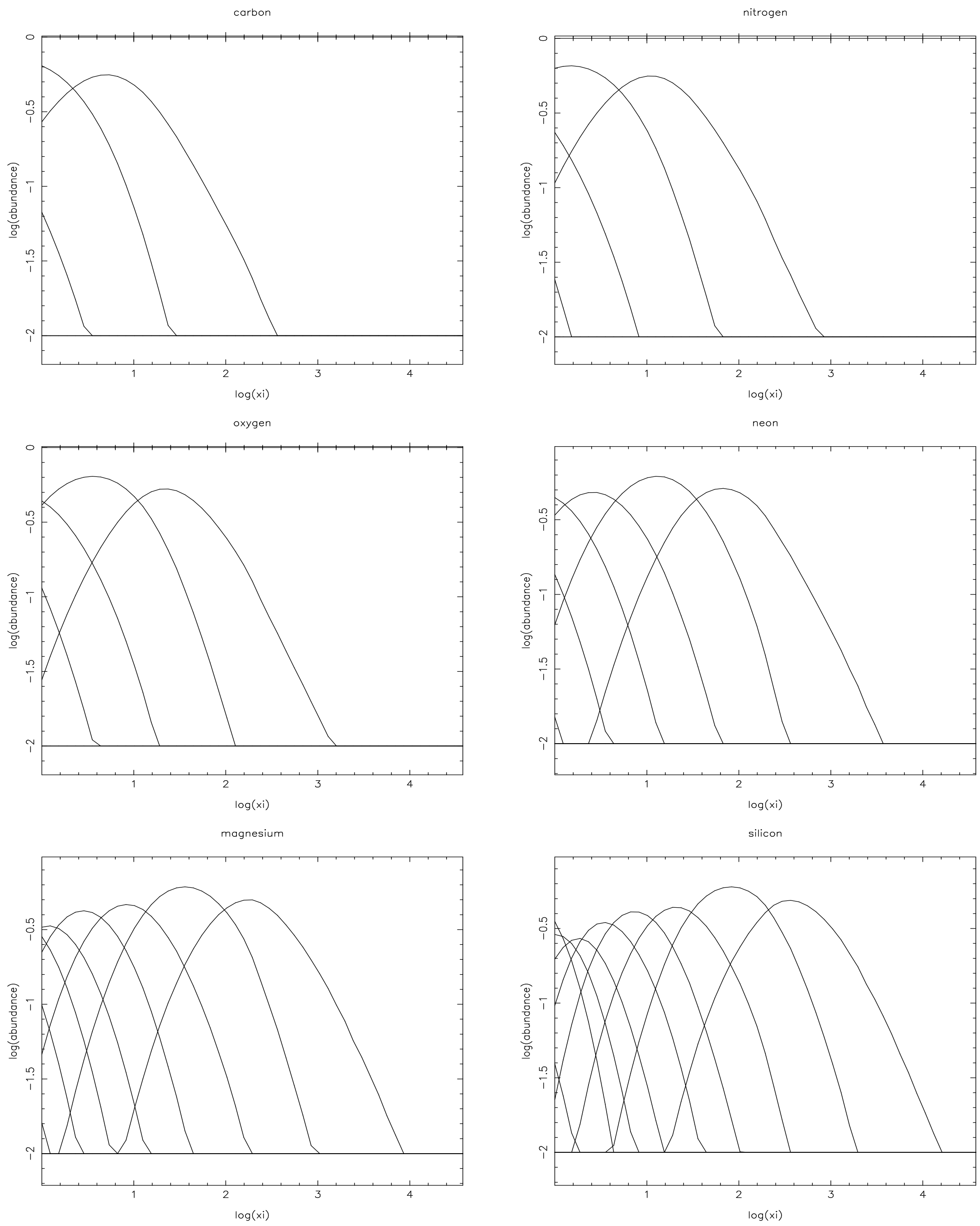

Fig. 13.— figure 13a

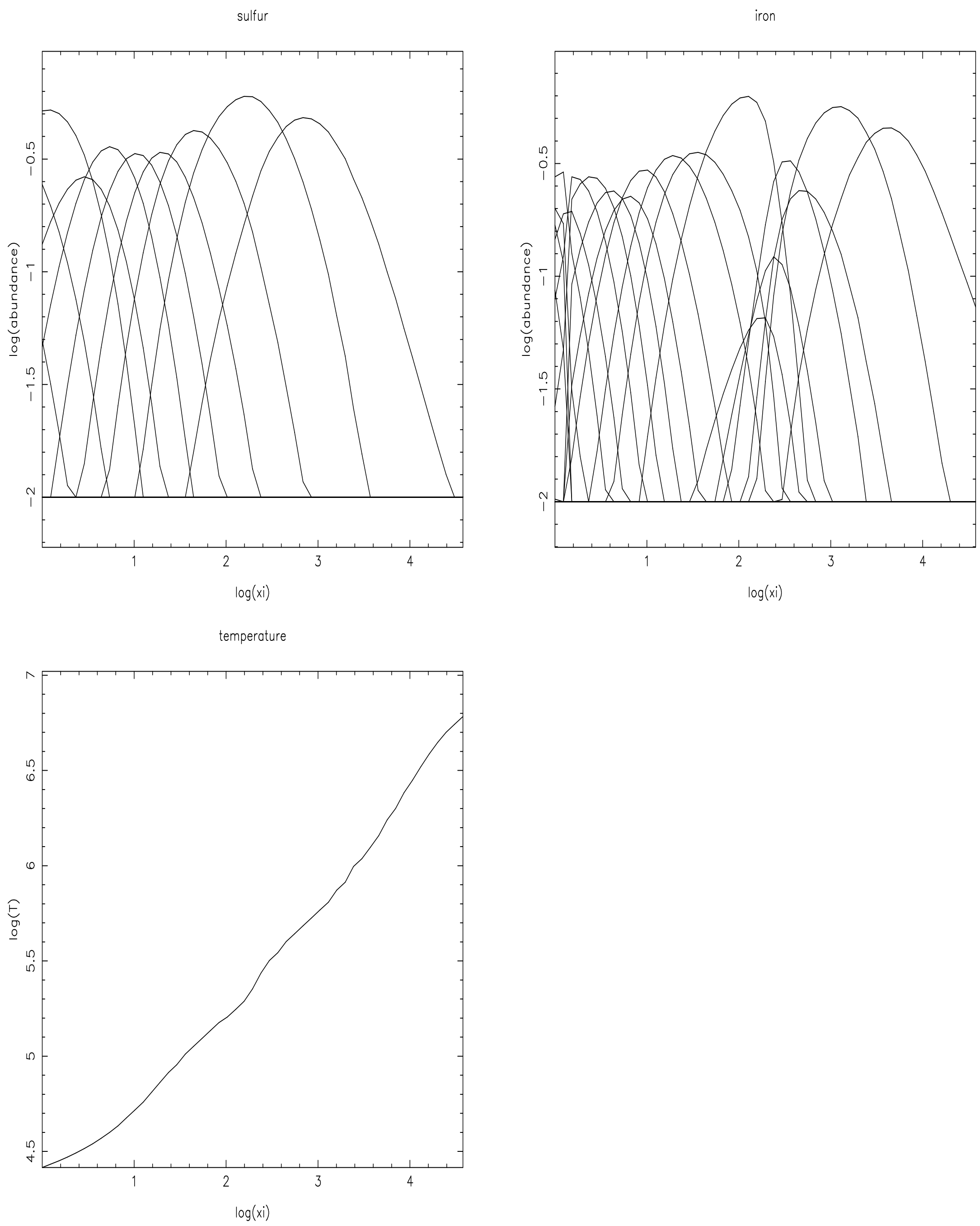

Fig. 13.— figure 13b

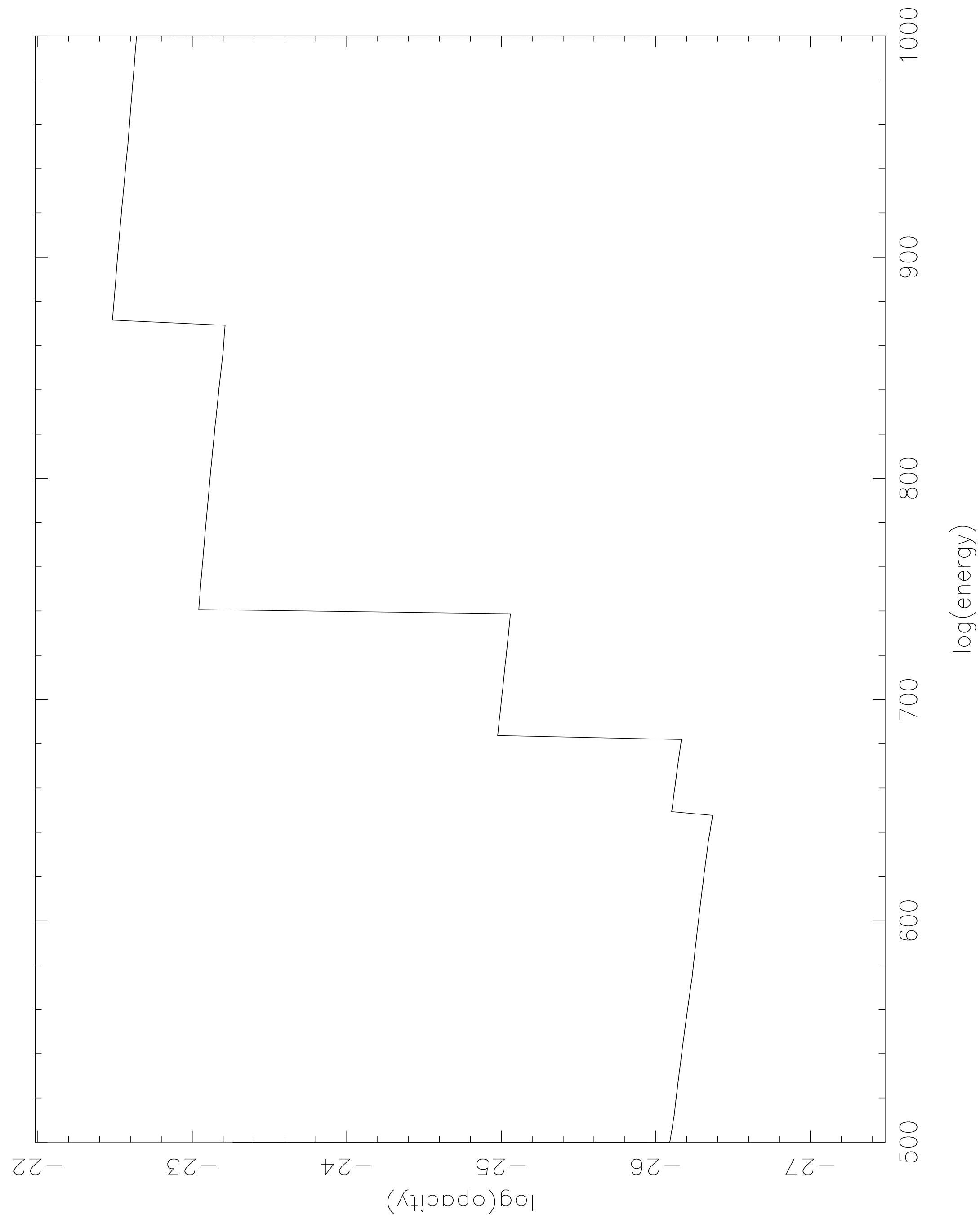

Fig. 14.— figure 14a

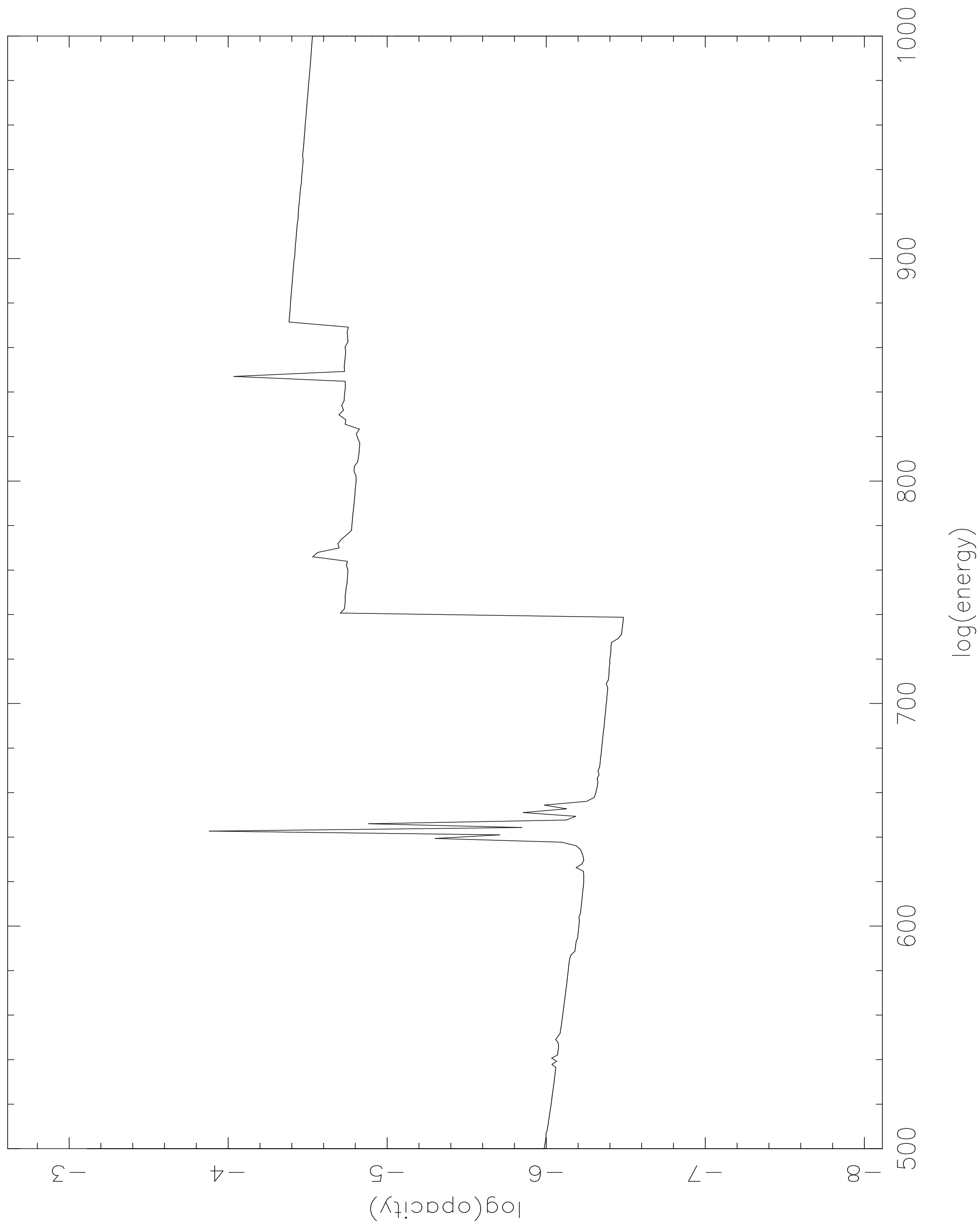

Fig. 14.— figure 14b

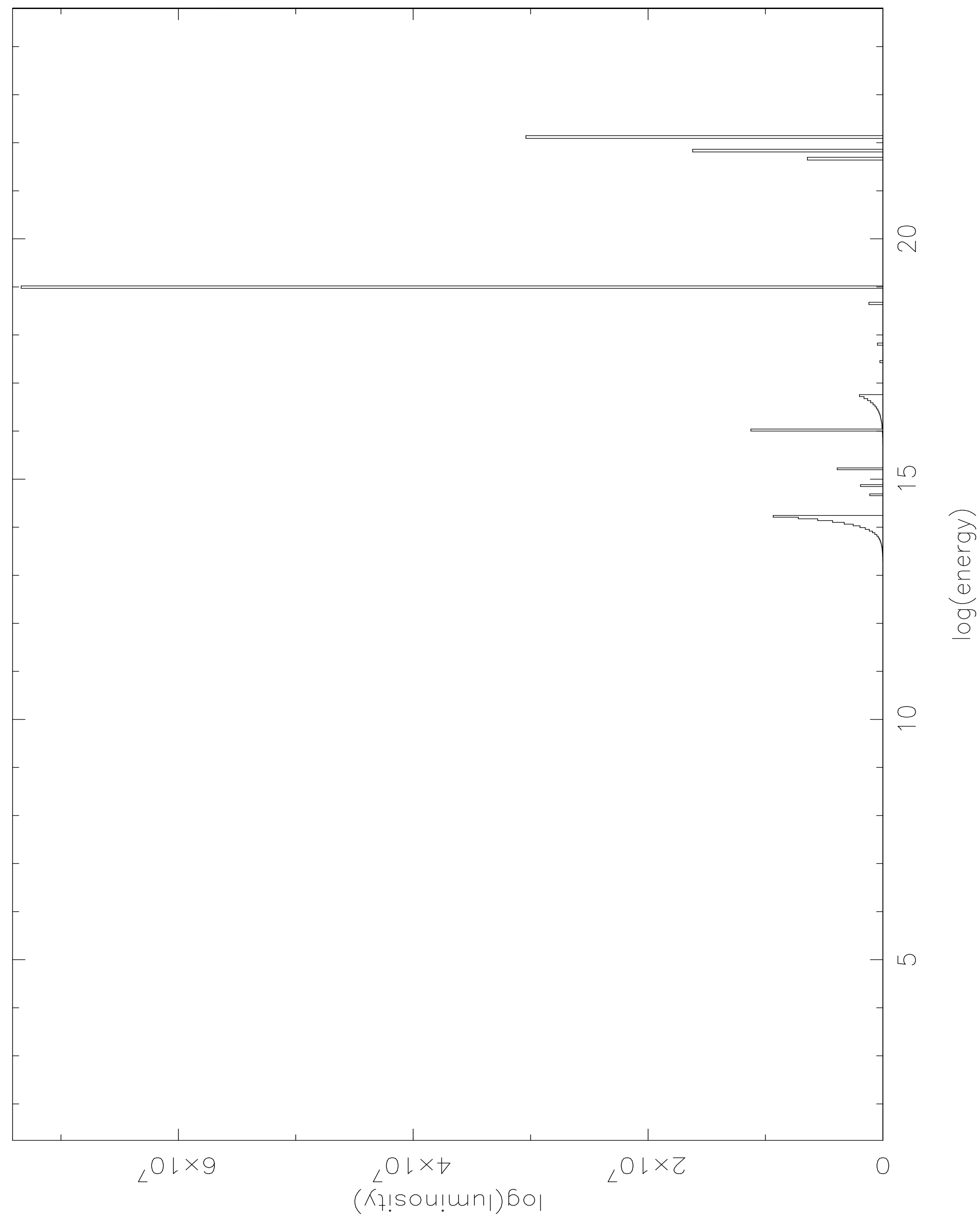

Fig. 15.— figure 15a

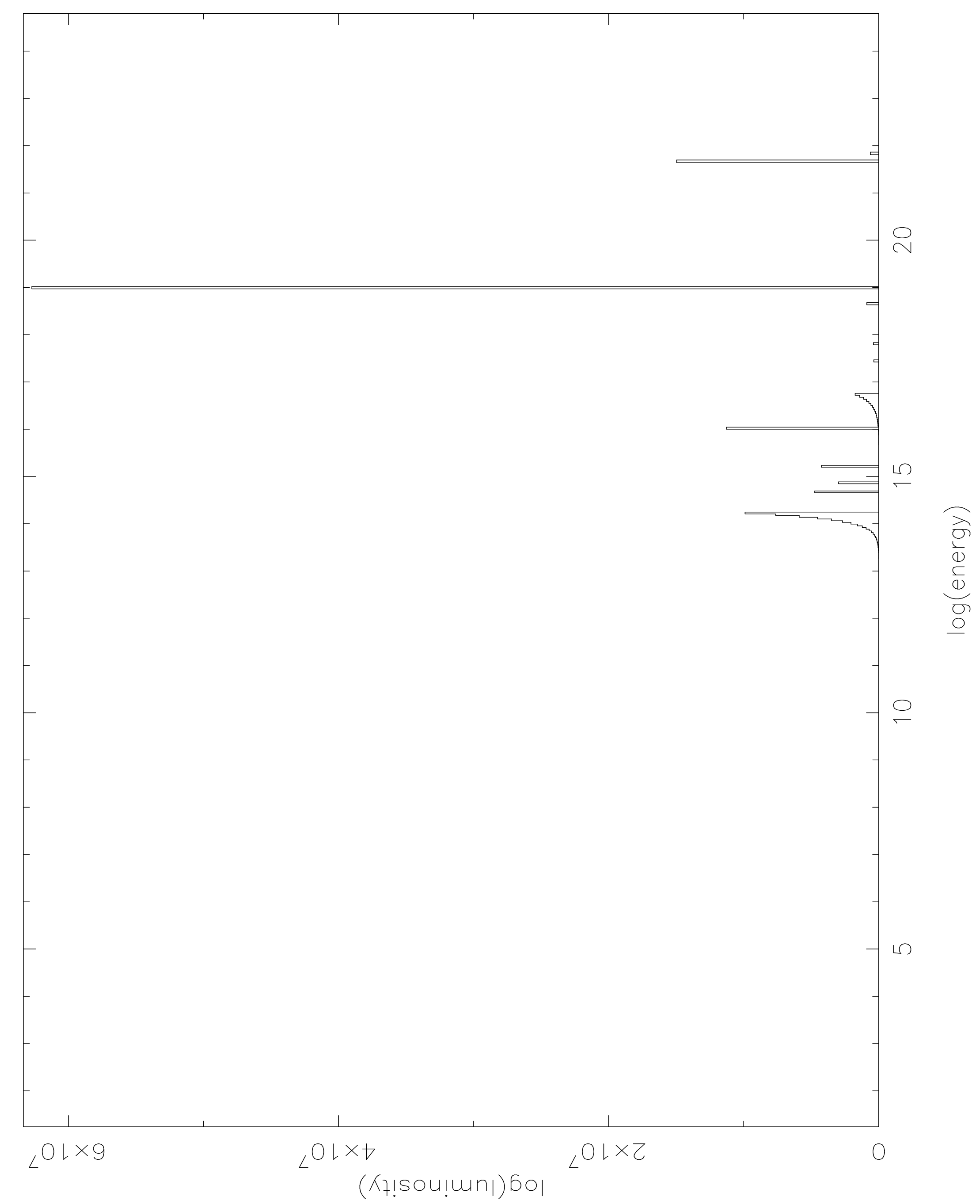

Fig. 15.— figure 15b

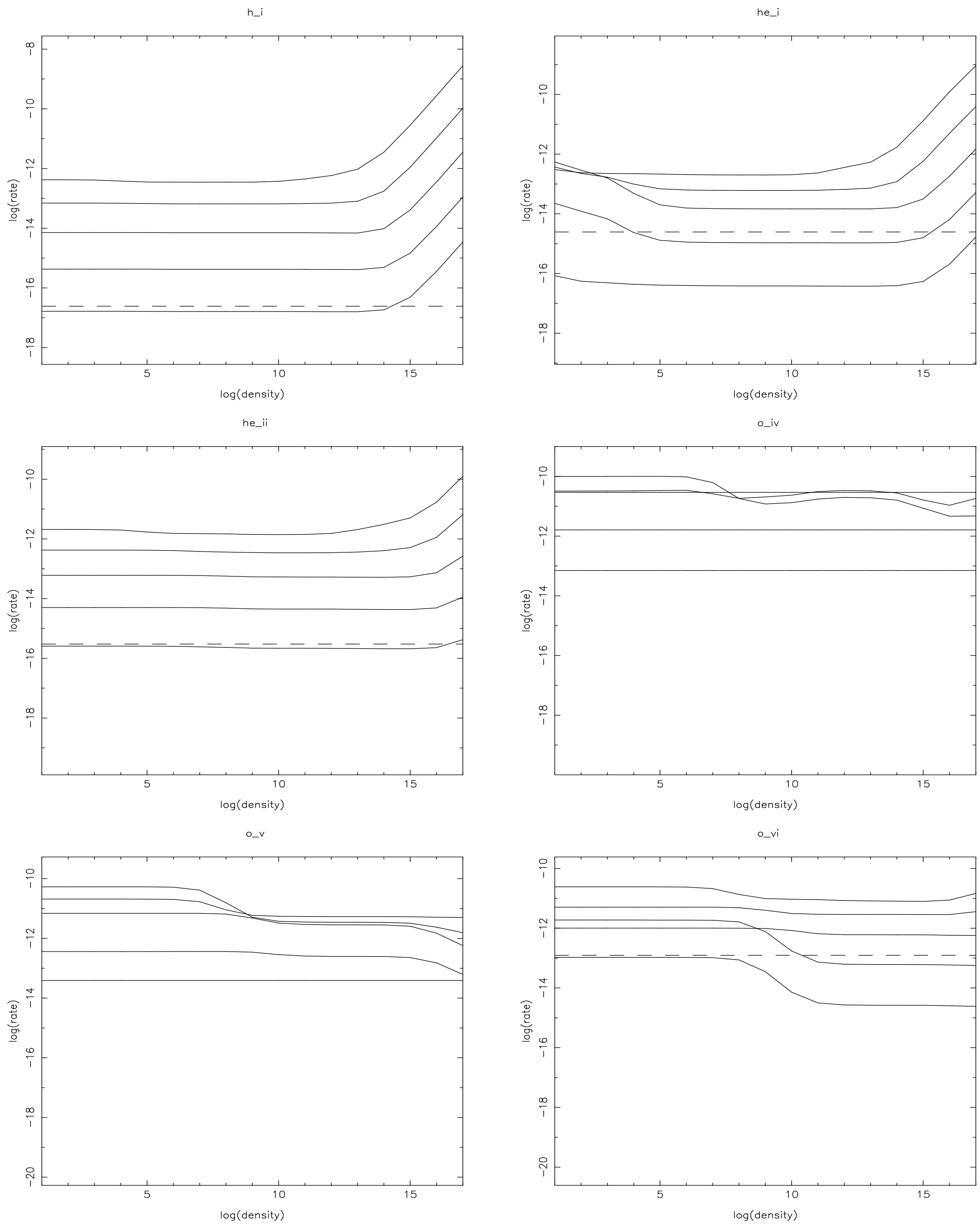

Fig. 16.— figure 16a

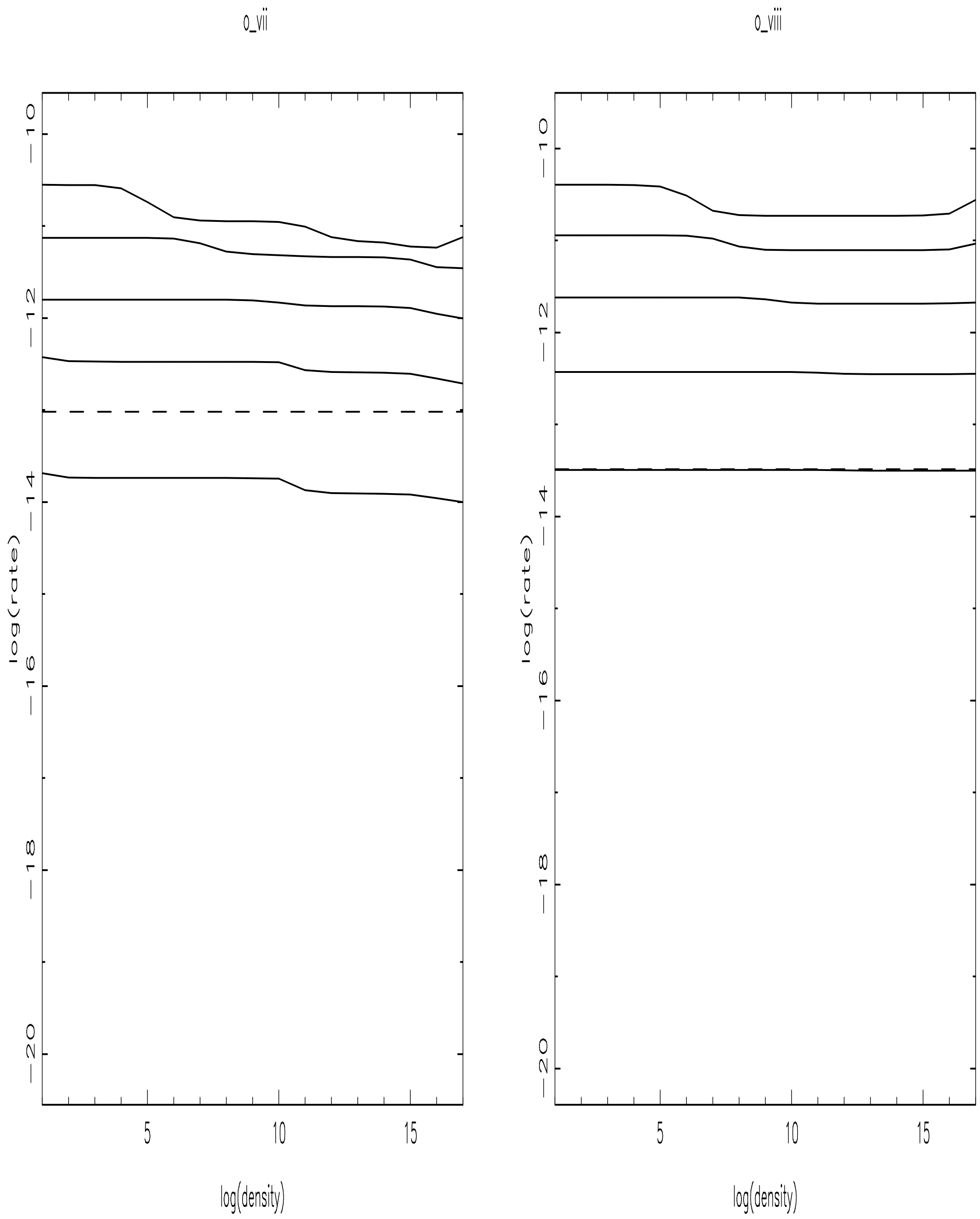

Fig. 16.— figure 16b

Chapter 17

Revisions

A. Version 2.1 (June 2000)

- Added 5 new input parameters vturbi, emult, critd, taumax, xeemin.

- Added artificial broadening of absorption lines due to turbulent velocity controlled by input parameter vturbi. If vturbi is less than the local thermal ion speed, then thermal Doppler broadening is used.

- Changed input spectrum in default parameter file to pow. Added error check for invalid input spectrum.

- Fixed error which displayed incorrect value of the constant pressure switch in output files.

- Several minor errors have been corrected in the some of the calculations of atomic rates: types 67, 51, and 70.

- Added printout of all line emissivities at each radial zone (in addition to the level populations) when the write_switch parameter is set to 1. These appear in the file named xout_detal2.fits.

- Modified the algorithm for calculating thermal equilibrium so that for temperatures less than 3030K the iteration procedure is disabled; any radius zone where the equilibrium solver finds a temperature less than this value will not calculate thermal equlibrium.

• Added 2 new output fits files to the standard output. xout_lines.fits contains the luminosities of the 100 strongest emission lines, and xout_cont.fits contains the continuum luminosities without the lines added (these two files can be combined by binning the lines suitably in order to make the contents of xout_spect1.fits).

• Added calculation and printout of LTE level populations to the quantities in xout_detail.fits.

• Improved treament of energy budget, and added printout of energy budget to xout_step.log.

B. Version 2.1a (December 2000)

• Fixed error in atomic rates affecting Fe XXI which caused recombination rates to be too large.

• Streamlined calculation of photoionization and recombination rate quadratures.

C. Version 2.1b (January 2001)

• Added printout of line and recombination cooling rates to printout in log file.

D. Version 2.1c (May 2001)

• Fixed error in database which resulted in too large emission in some iron Kβ fluorescence lines.

E. Version 2.1d (May 2001)

• Added feature which appends ion column densities in an additional extension to the xout_abund1.fits file.

F. Version 2.1e (June 2001)

• Extended the energy range up to 1 MeV and added relativisitic Compton heating and cooling.

• Fixed error in calculating threshold energies of some excited levels from type 53 data.

• Fixed error in subroutine which creates blackbody spectra which resulted in spuriously large fluxes at energies above 50 keV.

G. Version 2.1h (June 2002)

• Relativistic corrections to Compton heating and cooling have been added, using a procedure based on the work of Guilbert (1986).

• A new format for the storage of atomic data has been adopted, resulting in smaller files, and faster read times. This change should be transparent to the user, except for considerable speeding of data loading and program startup.

• A new algorithm for continuum transfer has been adopted, outward only transfer. This gives better energy conservation overall. A new column has been added in the log file output, labeled 'h-c', it tells the percent error in total energy conservation in the radiation field, i.e. total emitted - total absorbed.

• A new algorithm has been adopted for solving the statistical equilbrium, in place of LU decomposition. The algorithm involves an iterative solution to a simplified set of equations, and is described by Lucy (2001). This change should result in fast execution for models involving iron and other heavy elements, but will otherwise be transparent to the user.

• Various changes to atomic data, including increase in size of data file and addition of a new data type, for excitation of Fe XIX using data from Bhatia.

• Various minor inconsistencies and errors have been corrected, including spurious recombination emission to multiply excited levels.

• Standard output in the log file has been extended to include a list of the strongest absorption lines, and emission and absorption edges.

• Also, a new output file, xout_rrc1.fits, is created which contains a fits format list of the RRC strengths.

H. Version 2.1j (September 2003)

• New atomic data for iron K emission and absorption as described in Palmeri et al., 2004 A and A and references therein (see TK homepage for reprints).

• Added n=2-3 iron UTA absorption using data from FAC (Gu 2003).

• Fixed bug which limited length of spectrum file name to 8 characters.

• Fixed bug which allowed buffer containing ion fractions vs. xi to overflow when the number of spatial zones exceeded 1000. Now the limit on the number of spatial zones is 3000, and the code stops with a message when this is exceeded.

• Added accurate Voigt profile calculations for all lines in synthetic spectra.

• Fixed bug which limited length of spectrum file name to 8 characters.

I. Version 2.1k (May 2004)

• Added printouts of level opacitites, and level populations to final printout if the print switch is set to 2.

• Added a column to the printout of the file xout_detail.fits for the upp level index of each line.

• Repaired and streamlined the printing of the file xout_detail.fits.

• Added more informative statement when the code stops because the rate matrix overflows (ipmat too large).

• Added rate type 42: Auger decay

• Fixed arithmetic error which affected recombination rate calculation when kT E_{th}

• Added new data type (85) for photoionization resonances below threshold, along with new subroutine to calculate cross section (PEXS.f).

• Streamlined the photoionization rate calculation (phint53)

• More accurate treatment of line damping, Voigt profiles.

J. Version 2.1kn3 (April 2005)

• Two bugs were found in version 2.1k in the implementation of the Voigt function when calculating line absorption and the calculation of line broadening. The Voigt function bug affected primarily lines with small damping parameters, and resulted in non-fatal numerical errors in the xstar output absorption spectrum (INFs). When xstar was called as part of xstar2xspec this resulted in fatal errors because the cfitsio routines which read the xstar output could not interpret the INFs. The line broadening bug resulted in too large absorption line depths when turbulent broadening was important. Neither of these bugs affected the temperature, ionization balance or emission spectrum. The bugs have been repaired in version 2.1kn3.

- Version 2.1kn3 also has an added feature, which is the addition of ion-by-ion heating and cooling rates as extensions to the output file xout_abund1.fits.

- Also added is the capability to set the value of niter to a negative number, which allows the solution of charge conservation without solving thermal equilibrium. As before, if niter=0 then neither charge transfer nor thermal equlibrium is calculated.

K. Version 2.1kn4 (April 2005)

- Fixed an error which causes the wrong inital radius to be calculated when the constant pressure is option is chosen. Also changed the units label on the ionization parameter to remove inconsistency with constant pressure case.

- Minor changes in the radiation transfer algorithm.

L. Version 2.1kn5 (March (?) 2006)

- Fixed bug which affected high ionization models which included nickel. This caused segmentation faults, and was caused by an incorrect data type flag in the atomic data for He-like Ni.

- Changed step size algorithm to prevent stepping beyond the column density specified in the input. This will not be accurate for constant pressure clouds in which the temperature is changing rapidly.

M. Version 2.1kn6 (June 2006)

• Added the effect of photoionization and heating by line photons generated elsewhere in the cloud. These are photons which have already escaped the local region close to the point of emission.

• Changed the step size computation algorithm in order to account for the process of emission. That is, the step size is now is based on the length scale for significant change of both absorption and emission.

• Fixed several errors in the atomic database, notably affecting N-like ions. These affect some of the density sensitive lines in low ionization models.

• Update to this manual, in the chapter in the Physics of xstar, describing in more detail the radiation transfer algorithm.

N. Version 2.1kn7 (March 2007)

• A bug has been found affecting the intensities of the He-like forbidden lines from C, N, and O at high densities.

O. Version 2.1kn7 (December 2007)

The xstar database has been updated to take into account the iron M shell UTA data of Gu et al., 2006, 641, 1227. A revised database file for use with xstar21kn7 is available from the xstar website.

P. Version 2.1l

Version 2.1l represents an update to the atomic data which includes the iron and oxygen inner shell data which was presented in 2004 Ap. J. Supp. 155, 675, along with the line data for iron from Chianti 5 and features from version 2.1kn6. Recent updates include fixes to buge in the routines associated with xstar2xspec. These caused numerical problems on 64 bit machines, and also resulted in errors when large grids of models were run. Versions 2.1l, 2.1lnx, etc. have not yet been completely tested and so have not been put into the standard release.

Q. Version 2.1ln3

A bug has been found in version 2.1ln2 which affects the results in the paper 2004 Ap.J.Supp. 155 675. This is a bookkeeping error resulting in multiple-counting of the iron L shell cross section when calculating the cross sections for the 'third row' ions, Fe I – VIII. This makes a quantitative change to the results in figures 4a and 13a. That is, it affects opacity due to iron above approximately 1 keV, only for low ionization models (log(xi)¡0). These errors have been repaired in the version 2.1ln3, and repaired versions of the figures can be found on the xstar website.

R. Version 2.1ln4

Fixes to bugs in the routines associated with xstar2xspec. These caused numerical problems on 64 bit machines, and also resulted in errors when large grids of models were run (November 2007).

S. Version 2.1ln5

Incorporates the revised iron UTA data from Gu et al., 2006 (December 2007).

T. Version 2.1ln6

An update which implements strong typing to the fortran code. Function and results should be the same as previous versions.

U. Version 2.1ln7

Contains the dielectronic recombination rates for the ions of iron calculated by Badnell 2006 Ap. J. Lett. 651, 73. These result in a qualitative change to the ionization balance of iron for log(ξ)$\leq$1.

V. Version 2.1kn9/v2.1ln9 (November 2008)

Treatment of line profiles both in absorption and emission has been redone. Previously the profile function for each line was evaluated at the boundary of each energy bin. Now each energy bin contains the integrated line luminosity (or optical depth) within that bin. This will have a significant effect for lines which are narrower than the default bin spacing, which is approximately 350 km/s. This affects outputs in the binned spectrum in xout_spect1.fits.

W. Version v2.1ln10 (May 2009)

An error was found in the book keeping for some inner shell photoionization cross sections, resulting in double-counting in the opacity for some inner shell bound-free transions. These affect primarily low ionization models, and do not affect Fe or O. This has been fixed in this version of the code. This does not affect version 2.1kn9 and previous.

X. Version v2.1ln11 (May 2009)

An error was found in the zeroing of one of the arrays used for zeroing an important matrix which is used in calculating level populations. This led to spurious emission in some fluorescence lines from low-medium ionization species of elements other than O or Fe, all occuring in low-medium ionization models.

Y. Version v2.2.0 (November 2009)

This version includes the following added features: (i) Inclusion of all elements up to Z=30. The atomic data for the energy level structure of ions with 3 or more electrons for many of these are scaled hydrogenic and so the associated line emission must be treated with caution. (ii) Inclusion of two new input parameters: the radius exponent (radexp) and the number of continuum energy bind (ncn2). These are described in the chapter on input to xstar. (iii) Inclusion of the radiation scattered in resonance lines as a column in the output fits file xout_spect1.fits. This is provided in the same units of specific luminosity as the other columns. (iv) Use of a new algorithm for the multilevel calculation which is considerably faster and requires less storage. Hence smaller values of critf (even 0) can be accomodated for many problems. (v) The input parameter critf now refers to the fractional ion abundance (i.e. relative to the parent element) rather than the absolute (i.e. relative to H) ion abundance. (vi) Minor changes have been made to some of the output formats in the ascii file xout_step.lis. (vii) The atomic data for dielectronic recombination has been changed to incorporate the results from Badnell and coworkers (http://amdpp.phys.strath.ac.uk/tamoc/DATA/RR/) in place of the rates from and . This has quantitative effects on many of the results from xstar. Notable is the effect on the ionization balance of iron for ionization parameters in the range $0 \leq \log(\xi) \leq 2$, where the m-shell ions dominate, and where the new rates are greater than the previous ones by large factors.

Z. Version v2.2.1 (April 2010)

Fixes to bugs which affected the length of the name of the spectrum file used when the 'file' input option is specified, and which affected the operation of multi-pass runs.

REFERENCES

Aldrovandi, S., and Pequignot, D., 1973 Astr. Ap. 25, 137; 1976 47, 321

Arnaud, M., and Rothenflug, R., 1985, A. and A. Supp. 60 425

Arnaud M., and Raymond, J., 1992, Ap. J., 398, 394

Bambynek W., et al.1972, Rev. Mod. Phys. 44, 716

Barfield, W., Koontz, G. and Huebner, W., 1972, JQSRT, 12, 1409

Bautista, M., et al., 1997 Ap. J. Supp., 118, 259

Bautista, M., et al., 1998 Ap. J. in press

Bautista, M., et al., 1999 Ap. J. in press

Bautista, M., and Kallman, T., 2000, Ap. J. submitted

Butler, S., Heil, T.G., and Dalgarno, A., 1980, Ap. J., 241, 442

Cunto, et al., 1993, A and A, 275, L5

Dalgarno, A. and Butler, S. , 1978, Comments At. Mol. Phys., 7, 129

Dalgarno, A., Heil, T., and Butler, S., 1981, Ap. J., 245, 793

Davidson, K., and Netzer, H., 1979, Rev. Mod. Phys., 51, 715

Dere, K. P., et al., 1997, A and A S, 125, 149.

Field, G., and Steigman, G., 1971, Ap. J., 166, 59

George, I., et al., 1998, Ap. J. Supp. 114, 73

Gould, G. and Thakur, R., 1970, Ann. Phys., 61, 351

Grevesse, N., Noels, A., and Sauval, A., 1996, in "Cosmic Abundances" ASP Conference Series, 99, S. Holt and G. Sonneborn, eds.

Halpern, J., and Grindlay, J., 1980, Ap. J., 242, 1041

Hatchett, S., Buff, J., and McCray, R., 1976, Ap. J., 206, 847

Harrington, J.P., 1989, proceedings of the 131st symposium of the IAU held in Mexico City, Mexico, october 5-9, 1987. Ed S. Torres-Peimbert. Planetary Nebulae, 157-166

Hollenbach, D., and McKee, C., 1978, Ap. J. Supp., 41,555

Hummer, D., 1968, MNRAS, 138, 73

Hummer, D., et al., 1993, A and A, 279, 298

Hummer, D., and Rybicki, G., 1971, Ann. Rev. Astr. Ap., 9, 237

Illarionov, A., et al. 1979, Ap. J., 228, 279

Kallman, T., and McCray, R., 1980, Ap. J., 242, 615

Kaastra, J., and Mewe, R., 1989, A and A Supp., 97,443

Kallman, T., and McCray, R., 1982, Ap. J. Supp., 50, 49

Kallman 1983)

Karzas and Latter, 1966

Krolik, J., McKee, C., and Tarter, C. B., 1981, Ap. J., 249, 422

Kwan, J., and Krolik, J., 1981, Ap. J., 250, 478

London, R., 1979, Ap. J., 228, 8

Lotz, W., 1967, Ap. J. Supp., 14, 207

McCray, R., Wright, C., and Hatchett, S., 1977, Ap. J., 211, 29

Mendosa 1982)

Nahar, S., 1999, Ap. J. Supp., 120, 131

Nahar, S., 2000, Ap. J. in press

Seaton, M., et al., 1993, Rev. Mex. Astron. Astrofis., 23, 19

Osterbrock, D., 1974, Astrophysics of Gaseous Nebulae San Francisco: Feeman)

Press, W. et al., "Numerical Recipes"

Raymond and Smith 1986

Ross, R., Weaver, R., and McCray, R., 1978, Ap. J., 219, 292

Ross, R., 1979, Ap. J., 233, 334

Shull, J. M., 1979, Ap. J., 234, 761

Tarter, C. B., Tucker, W, and Salpeter, E., 1969, Ap. J., 156, 943

Tucker, W., and Koren, M., 1971, Ap. J., 168, 283

D. A. Verner and D. G. Yakovlev, 1995, A and A S, 109, 125

Weisheit, J., 1974, Ap. J., 190, 735

Zhang, H., and Sampson, D., 1987 Ap. J. Supp., 63, 487

Press - Numerical Recipes

This manuscript was prepared with the AAS LaTeX macros v4.0.